Pay by the Bit: An Information-Theoretic Metric for Collective Human Judgment

Tamsyn P. Waterhouse
Google, Inc.
345 Spear Street, 4th Floor
San Francisco, CA 94105-1689
tpw@google.com

ABSTRACT

We consider the problem of evaluating the performance of human contributors for tasks involving answering a series of questions, each of which has a single correct answer. The answers may not be known *a priori*.

We assert that the measure of a contributor's judgments is the amount by which having these judgments decreases the entropy of our discovering the answer. This quantity is the pointwise mutual information between the judgments and the answer.

The expected value of this metric is the mutual information between the contributor and the answer prior, which can be computed using only the prior and the conditional probabilities of the contributor's judgments given a correct answer, without knowing the answers themselves.

We also propose using multivariable information measures, such as conditional mutual information, to measure the interactions between contributors' judgments.

These metrics have a variety of applications. They can be used as a basis for contributor performance evaluation and incentives. They can be used to measure the efficiency of the judgment collection process. If the collection process allows assignment of contributors to questions, they can also be used to optimize this scheduling.

Author Keywords

human computation; information theory; crowdsourcing

ACM Classification Keywords

H.5.3. Information Interfaces and Presentation (e.g. HCI): Group and Organization Interfaces

INTRODUCTION

Background

Human computation, as defined by Quinn and Bederson [21] (*quo vide* for more discussion of it and related terms), is the performance of computational tasks by humans under the direction of a computer system. The term is sometimes confused with crowdsourcing, which encompasses a broader definition of tasks, performed by workers sourced openly online.

In this paper we restrict ourselves to a specific human computation paradigm we call *collective human judgment*: tasks phrased in the form of a question with discrete possible answers, such as the Freebase curation tasks described by Kochhar, Mazzocchi, and Paritosh [16]. Each question is presented independently to a number of contributors, and their judgments are synthesized to form an estimate of the correct answer.

Some examples will clarify this concept. A subjective or creative task such as writing ad copy [8], summarizing a written article, or rating others' work, sourced out to workers via a service such as Amazon Mechanical Turk, CrowdFlower, or oDesk, is crowdsourcing but is not collective human judgment. An objective computational task given to an established team of contributors, as in the case of Freebase data curation [16], is collective human judgment but is not crowdsourcing. Classification of a large data set using volunteers sourced online, such as the Galaxy Zoo project [19], is both crowdsourcing and collective human judgment. The present work is not applicable to the first example but is applicable to the other two.

Collective human judgment presents an interesting quality-control problem: How do we measure and manage the performance of human computers?

Salaried human computers are often under pressure to perform well, which usually means some combination of "faster" and "better". Definitions of these terms, and how they are weighted in evaluating workers' performance, can often be vague or capricious.

Crowdsourced workers, on the other hand, work online for low pay with minimal supervision. There is a financial incentive to perform the most work with the least effort, which can lead to laziness, cheating, or even adversarial behavior. Such behavior can be modeled and accommodated without difficulty in the processing of collective human judgments, but it still comes with a cost: Employers must pay workers, even for worthless work. Most solutions to this problem involve terminating (or blocking) low-quality workers and offering extra financial incentives to high-quality workers.

CSCW'13, February 23–27, 2013, San Antonio, Texas, USA.

Thus, there is a growing need for objective measurements of worker performance in human computation tasks, especially in cases where the workers are crowdsourced.

Aside from the need to be able to evaluate workers' contributions, the benefits of proactive management of human computers have been studied by Dow et al. [8]. Shaw, Horton, and Chen [24] explore the effect of several different incentive mechanisms on Mechanical Turk workers performing a website content analysis task. But without an objective metric for contributor performance, we have no way of knowing that we're incentivizing behavior that is actually beneficial.

Once answers are estimated, we could simply score workers by percent of answers correct. But this has serious shortcomings. In many cases, a good worker can produce judgments that are highly informative without ever being correct. For example, for classification problems over an ordinal scale, a well-meaning but biased worker can produce valuable output. If the distribution of classes is not uniform, a strategic spammer can often be correct simply by giving the most common answer every time. Problems over interval scales, for example the image-cropping task described by Welinder and Perona [30], can be resolved by modeling contributor estimates of the correct answer as discrete Gaussian distributions, with even a very good (accurate and unbiased) contributor having a very low probability of getting the answer exactly correct.

Many collective judgment resolution algorithms implicitly or explicitly include parameters representing contributors' skill. But such parameters are specific to their models and may not even be scalars, making it impossible to use them directly to score workers.

Ipeirotis, Provost, and Wang [15] introduce a data-based metric for contributor performance that calculates the hypothetical cost incurred by the employer of using that contributor's judgments exclusively. It depends on confusion matrices (and thus is restricted to classification problems) and takes a cost-of-errors matrix as input.

Paritosh [20] champions experimental reproducibility, specifically using Krippendorff's alpha [17] to measure the reliability of data from human contributors. Alpha is a descriptive statistic that measures aggregate agreement between contributors in a task. It's useful as a general measure of the statistical health of a task, but it doesn't say anything about the performance of individual contributors.

We propose a metric which we feel provides an objective and widely-applicable measurement of workers' contributions, in naturally-motivated units of information. Unlike Ipeirotis, Provost, and Wang's cost function, our metric is explicitly intended to be agnostic to the purpose for which the results are used: Workers are measured strictly by the information content of their judgments.[1]

The proposed metric also allows us to measure the redundancy of information obtained from multiple contributors during the judgment collection process. We can therefore measure the information efficiency of the process, and even tune the assignment of questions to contributors in order to increase the amount of information we obtain from a given number of judgments.

[1]However, we will see that we have the latitude to adjust incentives by attaching weights to different outcomes in the calculation of the metric.

Information Theory

The field of information theory [22, 23] deals with the mathematical aspects of the storage and communication of data.

The central concept of information theory is entropy, a measure of the expected (or asymptotic) minimal number of basic symbols (for example, bits) required to encode each of a sequence of outcomes of a random variable.

If X is a discrete random variable, the *self-information* of the outcome $X = x$ is defined to be

$$I(X = x) \equiv -\log P(X = x),$$

where the base of the logarithm determines the units in which the information is to be expressed. The most common choices are bits (base 2) or nats (base e), though familiar units from computing (decimal digits, bytes, etc.) can be used just as easily.

The *entropy* of X is defined to be the expected value of $I(X = x)$ over all x in the associated sample space, and is written as

$$H(X) \equiv E_X[I(X)] = \sum_x P(X = x)\left(-\log P(X = x)\right).$$

The definition of entropy is characterized, up to the choice of base, by four properties:

Continuity with respect to each probability $P(X = x)$.

Symmetry under exchange of outcomes $x \leftrightarrow x'$.

Maximization when all outcomes are equiprobable, with $P(X = x) = P(X = x')\forall x, x'$.

Additivity with respect to combining systems. The entropy of two independent variables X and Y is equal to $H(X) + H(Y)$.

Introductions to information theory are widely available, but Shannon's 1948 paper [22] remains both accessible and relevant today.

Information theory has been previously applied to topics in human-computer cooperation. For example, Card, Moran, and Newell [2] were among the first to describe human perception, cognition, and performance in computational terms, paraphrasing Hick's Law [14] as "to a first order of approximation, the response time of people is proportional to the information-theoretic entropy of the decision." Yang-Peláez and Flowers [34] discuss the information content of visual presentations of relational data. Chi and Mytkowicz [5] apply measures of information to search-and-retrieval tasks over the documents in a social tagging system. In the field of prediction markets, Hanson's logarithmic market scoring rules [11]

yield payoffs proportional to the relative entropy of the probability distributions before and after a trader changes the market probability [4]. In evolutionary programming, Card and Mohan [3] propose using mutual information in the evaluation of genetic algorithms.

Collective Human Judgment

We work under the regime of Bayesian inference, in which the posterior probability of a given correct answer, given a set of judgments, is proportional to its prior times the probability of obtaining those judgments given that answer.

Hence, in order to perform this inference, we need a model that provides the probability of obtaining a particular set of judgments for a given question. Assuming that the contributors' judgments are conditionally independent of each other, conditioned on the correct answer, then the model need only prescribe the probability distribution of each contributor's judgments given a correct answer. These models are typically parametrized by a vector of parameters for each contributor, but question-dependent parameters are also possible.

The choice of model depends strongly on the nature of the problem. Classification tasks have a common set of classes for all questions, and so a model that incorporates a contributor's possible confusion of one class with another is appropriate: This is the family of *confusion matrix* models. For tasks whose answers are numbers, an entirely different model based on continuous probability distributions may be appropriate. We'll discuss these issues in more detail later.

Although a model can provide all of the probabilities needed to compute a posterior answer distribution, we still need the values of the model's parameters. Simple approaches to inference typically make point estimates of these parameters, for example maximum-likelihood estimates obtained using the expectation-maximization algorithm, and use these point estimates to compute posteriors. A more rigorous approach is to perform marginal integration over the space of possible parameter values, but this requires more sophisticated numerical or analytical techniques and is more computationally intensive.

A common property of all of these techniques is that they are attempting to solve a hidden-data problem. If we knew all the answers, then we could easily determine the parameter likelihoods. Likewise, with known parameter likelihoods, we could determine answer posteriors. With no answers known, a bootstrapping process is necessary. Expectation-maximization addresses the problem by alternating between estimating answers from parameters and estimating parameters from the estimated answers, eventually converging to maximum-likelihood parameter values for the full hidden-data problem. Marginalization treats both answers and parameters as unknowns and integrates the answer posteriors over all possible parameter values, weighted by their likelihoods. In both regimes it's possible to inject "golden" data, in other words a subset of answers that are known *a priori*, to increase the accuracy of the inference.

Model and inference algorithm selection present many subtle issues that lie beyond the scope of the present work, and there is a great deal of relevant literature. Models for various task types are proposed by Whitehill et al. [31] (a logistic model linking a contributor skill parameter to a question difficulty parameter), Welinder and Perona [30] (simple models for several tasks including numeric ones), and Welinder et al. [29] (a probit model using vectors to represent problem domains and contributor skill domains). The expectation-maximization algorithm was introduced by Hartley [12], named by Dempster, Laird, and Rubin [7], and popularized by Dawid and Skene [6]. Numerical marginalization for high-dimensional inference problems generally follows the Markov chain Monte Carlo paradigm [13] (generating a sequence of samples from the domain of integration in which each sample is conditioned on the one before it) and may involve Gibbs sampling [10] (iteratively sampling each variable conditioned on all the others) or substitution sampling [27] (iterating between sampling the parameters and sampling the hidden data, in a stochastic analogue to expectation-maximization); see Gelfand and Smith [9] for an accessible introduction to these methods. Analytic approximations through variational Bayesian methods are also possible [1, 26] but are more mathematically challenging.

For our purposes, we will prefer simple models and use expectation-maximization to perform inference. We'll mention the limitations of this approach as we encounter them. The critical limitation is this: The metrics presented in this article are only as good as the model parameters that they're based on. With too little data, a poor choice of model, or an inaccurate resolution algorithm, our metrics will be inaccurate.

Notation

We use the term *task* to refer to a set of questions and the judgments collected for those questions.

Let $\mathcal{Q}$ be the set of questions. Let $\mathcal{H}$ be the set of contributors. Let $\mathcal{A}$ be the set of possible answers. Let $\mathcal{J}$ be the set of possible judgments. In many applications we have $\mathcal{J} = \mathcal{A}$, but this isn't strictly necessary. When $\mathcal{J} = \mathcal{A}$, we'll refer to the set simply as the answer space.

Let A be a random variable producing correct answers $a \in \mathcal{A}$. Let J and J' be random variables producing judgments $j, j' \in \mathcal{J}$ from two contributors $h, h' \in \mathcal{H}$. J and J' are assumed to depend on A and to be conditionally independent of one another given this dependence. Since each h corresponds to one random variable J, we will often represent contributors by J instead of h when there is no risk of confusion.

For brevity and to keep notation unambiguous, we'll reserve the subscript a to refer to answer variables A, and the subscripts j and j' to refer to judgment variables J (from contributor h) and J' (from contributor h') respectively.

Thus, let $p_a = P(A = a)$ be the prior probability that the correct answer of any question is a. Let $p_{a|j} = P(A = a|J = j)$ be the posterior probability that the correct answer to a question is a, given that the contributor represented by J gave judgment j. Other probabilities for variables A, J, and J' are defined analogously.

THE VALUE OF A JUDGMENT

Information Content of an Answer

Suppose first that we have no judgments for a given question. The probability distribution of its correct answer is simply p_a, the answer prior. The information content of a particular answer $A = a$ is

$$I(A = a) = -\log p_a.$$

The choice of base of the logarithm here simply determines the units (bits, nats, digits, etc.) in which the information is expressed. We'll use bits everywhere below.

$I(A = a)$ is often introduced in the literature as a measure of surprise: It measures how surprised we should be at finding the outcome a. For example, if the correct answer turns out to be a, we'd be more surprised by it with a prior $p_a = 0.1$ than we would with a prior $p_a = 0.5$.

We could also write the information as $I(A)$, a random variable that depends on A. Its expected value is the *entropy* of A:

$$H(A) \equiv E_A[I(A)] = -\sum_a p_a \log p_a$$

Information Content of a Judgment

Suppose that we have a single judgment $J = j$ (from contributor h). Now the information content of the outcome $A = a$, conditioned on this information, is

$$I(A = a|J = j) = -\log p_{a|j},$$

and the expected value of $I(A|J)$ over all combinations of x and y is the *conditional entropy* of A given J,

$$H(A|J) \equiv E_{A,J}[I(A|J)] = -\sum_{a,j} p_{a,j} \log p_{a|j}.$$

Given a question $q \in \mathcal{Q}$, a judgment $j \in \mathcal{J}$, and the correct answer $a \in \mathcal{A}$, the information given to us by J is the amount by which the information content of the outcome $A = a$ is decreased by our knowledge of J: in other words, how much less surprising the outcome is when we have $J = j$. This quantity is

$$\Delta I_{a,j} \equiv I(A = a) - I(A = a|J = j) = \log \frac{p_{a|j}}{p_a}, \quad (1)$$

the *pointwise mutual information* of the outcomes $A = a$ and $J = j$.

We propose $\Delta I_{a,j}$ as a measure of the value of a single judgment from a contributor. If the judgment makes us more likely to believe that the answer is a, then the value of the judgment is positive; if it makes us less likely to believe that the answer is a, then its value is negative.

To compute $\Delta I_{a,j}$ for a single question, we must know the answer a to that question. In practice, we find a using a judgment resolution algorithm applied to a set of contributor judgments. Although knowing a is the goal of judgment resolution, we can still compute a contributor's aggregate value per question without it, simply taking the expected value of $\Delta I_{a,j}$ over all values of A and J. This is

$$E_{A,J}[\Delta I_{a,j}] = H(A) - H(A|J) \equiv I(A; J),$$

the *mutual information* of A and J, a measure of the amount of information the two random variables share. We can expand this as

$$I(A; J) = \sum_{a,j} p_{a,j} \log \frac{p_{a|j}}{p_a}. \quad (2)$$

Although ΔI can be negative, $I(A; J)$ is always non-negative: No contributor can have a negative expected value, because any behavior (even adversarial) that statistically discriminates between answers is informative to us, and behavior that fails to discriminate is of zero value.

a	j	$p_{a,j}$	$p_{a\|j}$	$I(A = a\|J = j)$	$\Delta I_{a,j}$
a_1	a_1	0.5	$\frac{2}{3}$	0.58 bits	0.42 bits
a_2	a_1	0.25	$\frac{1}{3}$	1.58 bits	-0.58 bits
a_2	a_2	0.25	1	0	1 bit

Table 1. Outcomes for the classification example.

Example

Say we have a task involving classifying objects q into two classes, a_1 and a_2, with uniform priors $p_{a_1} = p_{a_2} = 0.5$, so that $I(A = a_1) = I(A = a_2) = 1$ bit, and $\mathcal{J} = \mathcal{A}$.

Consider a contributor, represented by a random variable J, who always identifies a_1 objects correctly but misidentifies a_2 as a_1 half the time.

The possible outcomes for the random variables are listed in table 1.

This contributor is capable of giving a judgment that reduces our knowledge of the correct answer, with $\Delta I_{a_2,a_1} < 0$. However, the expected value per judgment from the contributor is positive: $I(A; J) = E_{A,J}[\Delta I_{a,j}] \approx 0.31$ bits.

Worker Incentives

If contributor rewards are linked to mutual information $I(A; J)$, then we are rewarding contributors for the degree to which they help us arrive at answers to our questions. If the computation task results in valid estimates of the answers, then we can say a few things about using $I(A; J)$ in a worker incentive mechanism.

It's important to note that $I(A; J)$ doesn't reward *honest* behavior so much as it rewards *consistent* behavior. A biased, or even adversarial, contributor who consistently swaps classes A and B in a classification problem, or one who consistently underestimates answers by 10 units in a numeric problem, is still providing us with useful information. In the former case we can reverse the swap, and in the latter case we can simply add 10 to the contributor's judgment, in order to get a good estimate of the correct answer.

There is one important caveat, however: We must have good estimates of the contributor parameters in order to have a

good estimate of $I(A;J)$. This is a particular hazard if the number of judgments is small and the resolution algorithm uses point estimates of parameters rather than marginal integration. This is analogous to the problem of guessing the ratio of colors of balls in an urn from a small sample: If I draw one red ball, the maximum-likelihood estimate is that all the balls are red.

It's also critical that contributors work independently of one another, because any resolution algorithm, even one incorporating golden data, can be misled by workers colluding to agree on answers to questions.

The point is that our ability to estimate worker performance is only as good as our ability to estimate answers. As we discussed in the Introduction, the resolution of answers in collective human judgment problems is a fertile area of research, beyond the scope of this article, and involves choosing both an appropriate statistical model for contributor behavior and an appropriate means of performing statistical inference over the model.

COMBINING JUDGMENTS

We typically have several judgments per question, from different contributors. The goal of judgment resolution can be expressed as minimizing $H(A|J,J')$, or in the general case, minimizing $H(A|J,\ldots)$. To do so efficiently, we would like to choose contributors so as to minimize the redundant information among the set of contributors assigned to each question.

Suppose we start with one judgment, $J=j$. If we then get a second judgment $J'=j'$, and finally discover the correct answer to be $A=a$, the difference in information content of the correct answer made by having the second judgment (compared to having just the first judgment) is the *pointwise conditional mutual information*

$$\begin{aligned}\Delta I_{a,j'|j} &\equiv I(A=a|J=j) - I(A=a|J=j,J'=j') \\ &= \log\frac{p_{a|j,j'}}{p_{a|j}}. \end{aligned} \tag{3}$$

Below, we'll consider two situations: assigning a second contributor represented by random variable J' to a question after receiving a judgment from a first contributor represented by J (the sequential case), and assigning both contributors to the same question before either gives a judgment (the simultaneous case).

This section tacitly assumes that we have a good way to estimate probabilities as we go. This requires us to work from some existing data, so we can't use these methods for scheduling questions until we have sufficient data to make initial estimates of model parameters. What constitutes "sufficient data" depends entirely on the data and on the resolution algorithm, and these estimates can be refined by the resolution algorithm as more data is collected.

For example, one could break the judgment collection process into N batches of judgments. Contributors are assigned to questions at random for the first batch, after which the resolution algorithm estimates answers and contributor metrics and then assigns questions to contributors optimally for the next batch, and so on. After all N batches have been collected, the resolution algorithm computes final estimates of answers and metrics.

Sequentially

The expected information gained from the second judgment, conditioned on the known first judgment $J=j$, is

$$E_{A,J'|J=j}[\Delta I_{a,j'|j}] = \sum_{a,j'} p_{a,j'|j}\log\frac{p_{a|j,j'}}{p_{a|j}}, \tag{4}$$

which is similar in form to equation (2).

This expression allows us to compute the expected value of $\Delta I_{a,j'|j}$ for each choice of second contributor J' and to make optimal assignments of contributors to questions, if our judgment collection system allows us to make such scheduling decisions.

Simultaneously

If we must choose contributors for a question before getting judgments, we take the expectation of equation (3) over all three random variables rather than just A and J'. The result is the *conditional mutual information*

$$I(A;J'|J) = E_{A,J,J'}[\Delta I_{a,j'|j}] = \sum_{a,j,j'} p_{a,j,j'}\log\frac{p_{a|j,j'}}{p_{a|j}}, \tag{5}$$

which is the expected change in information due to receiving one judgment from J' when we already have one judgment from J.

Two contributors are on average at least as good as one: The total information we gain from the two judgments is

$$I(A;J,J') = I(A;J) + I(A;J'|J) \geq I(A;J),$$

since $I(A;J'|J)\geq 0$. In fact, we can write

$$\begin{aligned} I(A;J,J') &= I(A;J) + I(A;J'|J) \\ &= I(A;J) + I(A;J') - I(A;J;J'), \end{aligned}$$

where $I(A;J;J')$ is the *multivariate mutual information* of A, J, and J'.[2] Here it quantifies the difference between the information given by two contributors and the sum of their individual contributions, since we have

$$I(A;J;J') = I(A;J) + I(A;J') - I(A;J,J').$$

$I(A;J;J')$ is a measure of the redundant information between the two contributors. Because contributors are responsible only for their own judgments, it might not be fair to reward a contributor's judgments based on their statistical interaction with another contributor's judgments. However, we can use this as a measure of inefficiency in our system. We receive an amount of information equal to $I(A;J)+I(A;J')$,

[2]Unfortunately, the same notation is used for the interaction information, which differs from multivariate mutual information by a factor of $(-1)^n$, where n is the number of variables involved.

but some is redundant and only the amount $I(A; J, J')$ is useful to us. The overlap $I(A; J; J')$ is the amount wasted.[3]

More than Two Judgments

Generalizing to higher-order, the relevant quantities are the combined mutual information

$$I(A; J_1, \ldots, J_k),$$

which gives the total information we get from k judgments, the mutual information conditioned on a set of known judgments $\{J_1 = j_1, \ldots, J_{k-1} = j_{k-1}\}$,

$$E_{A,J_k|J_1=j_1,\ldots,J_{k-1}=j_{k-1}}[\Delta I_{a,j_k|j_1,\ldots,j_{k-1}}] = \sum_{a,j_k} p_{a,j_k|j_1,\ldots,j_{k-1}} \log \frac{p_{a|j_1,\ldots,j_k}}{p_{a|j_1,\ldots,j_{k-1}}}, \quad (6)$$

which gives the expected increase in information from the kth judgment with the previous $k-1$ judgments already known, and the conditional mutual information

$$I(A; J_k|J_1, \ldots, J_{k-1}),$$

which gives the increase in information we get from the kth judgment with the previous $k-1$ judgments unknown. Higher-order expressions of multivariate mutual information are possible but not germane to our work: We are instead interested in quantities like

$$\left[\sum_{i=1}^{k} I(A; J_i)\right] - I(A; J_1, \ldots, J_k)$$

for measuring information overlap.

PRACTICAL CALCULATION

The equations we derived in the previous section are rather abstract, phrased as sums involving quantities like $p_{a|j}$. In this section we'll discuss how to compute these quantities in the context of real-world human judgment tasks.

Regardless of the type of task, we generally have a statistical model for questions and contributor behavior which defines probabilities p_a (answer priors) and $p_{j|a}$ and $p_{j'|a}$ (probability of a judgment given an answer). Other probabilities, such as $p_{a|j}$ (probability of an answer given a judgment), must be computed from these.

Pointwise mutual information, equation (1), is

$$\Delta I_{a,j} = \log \frac{p_{a|j}}{p_a} = \log \frac{p_{j|a}}{\sum_{a'} p_{a'} p_{j|a'}}. \quad (7)$$

Mutual information, equation (2), is

$$I(A; J) = \sum_{a,j} p_{a,j} \Delta I_{a,j} = \sum_{a,j} p_a p_{j|a} \log \frac{p_{j|a}}{\sum_{a'} p_{a'} p_{j|a'}}. \quad (8)$$

[3] $I(A; J; J')$ can in theory be positive or negative, but in most cases it's positive, representing redundant information. Cases in which it's negative represent information present in the combination of two judgments in excess of the sum of their individual contributions. Such cases can be contrived but have not been observed in our experiments.

To compute probabilities involving more than one judgment, we must assume conditional independence of judgments given the correct answer. This means, for example, that $p_{j'|a,j} = p_{j'|a}$.

Pointwise conditional mutual information, equation (3), is

$$\Delta I_{a,j'|j} = \log \frac{p_{a|j,j'}}{p_{a|j}} = \log \left(p_{j'|a} \frac{\sum_{a'} p_{a'} p_{j|a'}}{\sum_{a'} p_{a'} p_{j|a'} p_{j'|a'}} \right). \quad (9)$$

Mutual information conditioned on one point $J = j$, equation (4), is

$$E_{A,J'|J=j}[\Delta I_{a,j'|j}] = \sum_{a,j'} p_{a,j'|j} \Delta I_{a,j'|j} = \sum_{a,j'} \frac{p_a p_{j|a} p_{j'|a}}{\sum_{a'} p_{a'} p_{j|a'}} \log \left(p_{j'|a} \frac{\sum_{a'} p_{a'} p_{j|a'}}{\sum_{a'} p_{a'} p_{j|a'} p_{j'|a'}} \right). \quad (10)$$

Conditional mutual information, equation (5), is

$$I(A; J'|J) = \sum_{a,j,j'} p_{a,j,j'} \Delta I_{a,j'|j} = \sum_{a,j,j'} p_a p_{j|a} p_{j'|a} \log \left(p_{j'|a} \frac{\sum_{a'} p_{a'} p_{j|a'}}{\sum_{a'} p_{a'} p_{j|a'} p_{j'|a'}} \right). \quad (11)$$

Although these expressions may seem intimidating, the arguments of the logarithms are Bayes' Rule calculations that must be performed by the resolution algorithm, so that little additional work is required to compute these information-theoretic quantities.

Combined mutual information is then

$$I(A; J, J') = I(A; J) + I(A; J'|J),$$

and multivariate mutual information is

$$I(A; J; J') = I(A; J') - I(A; J'|J).$$

The exact nature of the statistical model depends in part on the type of problem we're trying to solve. Below we'll examine three different types of tasks, which we'll refer to as *classification*, *search*, and *numeric*.

Classification Tasks

Classification tasks are the bread and butter of the human computation literature. Such tasks ask contributors to bin objects into a predetermined set of classes or to determine for each object whether a given proposition is true. A good example of a classification task is Galaxy Zoo [19], which uses volunteers to classify galaxy morphology from photographs.

Although other models are possible, contributor behavior for classification problems can usually be modeled well with confusion matrices, $p_{j|a} \equiv \pi_{a,j}$, and the small size of the answer space relative to the number of questions allows us to make good estimates of the class priors p_a. In other words, we have a model that contains a confusion matrix $\pi_{a,j}$ for each contributor and a prior p_a for each class.

As long as our model provides $p_{j|a}$ and p_a, we can evaluate equations (7) to (11) directly using these quantities.

However, due to the multiple sums over the answer and judgment spaces that appear in the above equations, a few words on computational complexity are appropriate here.

Calculating contributor performance goes hand-in-hand with the main problem of turning human judgments into estimates of correct answers to questions. Our metrics can be computed at the same time as these estimates.

In the following section's examples, we'll assume for simplicity a judgment resolution algorithm which makes pointwise estimates of model parameters. Although this is a common simplification in the literature and in practice, full statistical inference involves marginal integration over the space of possible parameter values, for example using Markov chain Monte Carlo methods (see Walsh's notes [28] for an introduction). Instead of computing, say, $I(A; J'|J)$ using point estimates of parameters like p_a and $p_{j|a}$, we marginalize by integrating $I(A; J'|J) = \int_{\boldsymbol{\alpha}} I(A_{\boldsymbol{\alpha}}; J'_{\boldsymbol{\alpha}}|J_{\boldsymbol{\alpha}}) d\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ represents the full set of parameters governing the model.

Because these numerical methods are computationally intensive, we should consider how much additional burden our metrics would impose in such cases.

Let $m = |\mathcal{H}|$ be the number of contributors, $s = |\mathcal{A}|$ be the number of possible answers, and $t = |\mathcal{J}|$ be the number of possible judgments. Computing first-order quantities $I(A; J)$ for all contributors using equation (8) has run time $O(m \cdot s \cdot t)$, and computing a second-order quantity such as $I(A; J, J')$ for all contributor pairs using equation (11) has run time $O(m^2 \cdot s \cdot t^2)$. Higher-order quantities become challenging, though: In general, computing $I(A; J_1, \ldots, J_k)$ for all contributor k-sets has run time $O(m^k \cdot s \cdot t^k)$.

By comparison, estimating answers to all questions in a task from a single set of model parameter values can be accomplished in run time proportional to the total number of judgments received, which is typically of order 10 times the number of questions.

Search Tasks

Classification problems are characterized by a small answer space shared by all questions and sufficient data to estimate priors p_a and confusion matrix elements $\pi_{a,j}$.

We'll refer to problems that don't satisfy these conditions as "search problems", because unlike classification problems, we have little prior knowledge of what the answer could be: Our contributors are searching for it.

Search problems can have very large answer spaces. For example, the answer space for the question "Find the URI for company X's investor relations page" is the set of all valid URIs.

Problems with large answer spaces have large entropies, naturally incorporating the increased difficulty of searching over these spaces. The set of all Freebase MIDs (canonical topic identifiers), for example, presently numbers in excess of 20 million entities, so $H(A) \sim 24$ bits for problems over this space.

Large answer spaces aren't strictly necessary for a problem to fall into the search category, though: A typical multiple-choice exam also has the property that answer A on question 1 is unrelated to answer A on question 2, so confusion matrix models are inappropriate.

As we noted above, our metrics require a run time which is polynomial in the size of the answer spaces. For large answer spaces this becomes unviable. Fortunately, the same challenge is faced by the judgment resolution algorithm, and its solution can be ours too.

Without enough information to model contributor behavior using confusion matrices (or with answer spaces too large for confusion matrices to be computable), we usually use models with agnostic answer priors and a single value π that gives the probability of a contributor answering a question correctly. π may depend on the contributor, the question, or both; see Whitehill et al. [31] for an example of the latter case.

That is, we generally have $\mathcal{J} = \mathcal{A}$ and uniform answer priors $p_a = \frac{1}{s}$ (although this can change if we have information beyond just the contributors' judgments), where as above we let $s = |\mathcal{A}|$. Our conditional probabilities are

$$p_{j|a} = \pi \delta_{a,j} + \frac{1 - \pi}{s - 1}(1 - \delta_{a,j}),$$

where $\delta_{a,j}$ is the Kronecker delta. This means that with probability π, the contributor answers the question correctly, and with probability $1 - \pi$, the contributor gives a judgment chosen randomly and uniformly from among the incorrect answers.

All of the equations above now simplify (in terms of computational, if not typographical, complexity). The normalization term in equation (7) is

$$\sum_a p_a p_{j|a} = \sum_a \frac{1}{s} \left[\pi \delta_{a,j} + \frac{1 - \pi}{s - 1}(1 - \delta_{a,j}) \right] = \frac{1}{s}.$$

Conditional probability is

$$p_{a|j} = \frac{p_a p_{j|a}}{\sum_{a'} p_{a'} p_{j|a'}} = p_{j|a}.$$

Mutual information is therefore

$$I(A; J) = \pi \log(s\pi) + (1 - \pi) \log \left(s \frac{1 - \pi}{s - 1} \right).$$

Note that $I(A; J) \sim \pi \log s = \pi \cdot H(A)$ as $s \to \infty$. In figure 1, we plot $\frac{I(A;J)}{H(A)}$ for various values of s.

We can also compute higher-order quantities. For example, letting π be the probability-correct parameter for J and π' be

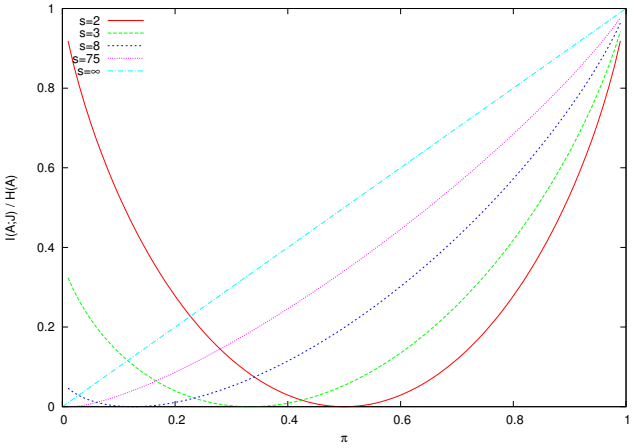

Figure 1. $\frac{I(A;J)}{H(A)}$ as a function of the probability-correct parameter π, for various sizes s of the answer space.

the probability-correct parameter for J', we have

$$\sum_a p_a p_{j|a} p_{j'|a} = \begin{cases} \frac{1}{s}\left[\pi\pi' + \frac{(1-\pi)(1-\pi')}{s-1}\right] & \text{if } j = j' \\ \frac{1}{s}\left[\frac{\pi + \pi' - 2\pi\pi'}{s-1} + \frac{(1-\pi)(1-\pi')(s-2)}{(s-1)^2}\right] & \text{if } j \neq j' \end{cases}.$$

For large s, we can approximate this as

$$\sum_a p_a p_{j|a} p_{j'|a} \approx \begin{cases} \frac{\pi\pi'}{s} & \text{if } j = j' \\ \frac{1-\pi\pi'}{s(s-1)} & \text{if } j \neq j' \end{cases}.$$

This simplification happens because for a large answer space, $j = j' \neq a$ occurs very rarely, so the $j = j'$ case is dominated by $j = j' = a$. This is another example of the assumption of contributor independence: Two contributors' wrong answers are assumed to be uncorrelated.

However, higher-order quantities are less useful for search problems than for classification problems, because with lower-dimension model parameters for each contributor, there is less to differentiate contributors from each other and thus less to be gained from attempting to optimize scheduling. In the case of just one parameter π per contributor, contributor skill is literally one-dimensional.

In any event, what matters is that for search problems, we can eliminate the sums in equations (7) to (11) to produce computationally simple quantities.

Numeric Tasks

A third class of human judgment problem is that of estimating a numerical value. Such tasks differ from search tasks in that the answer space is interval rather than categorical: A judgment close to the correct answer can still be informative. Welinder and Perona [30], for example, consider the problem of determining a bounding rectangle around a desired object in an image. Human contributors were shown a series of images, each of which contained a bird, and asked to draw a "snug" bounding box around the bird in each image. The judgments and answers to these questions are integer 4-tuples representing the x and y coordinates of each edge of the bounding box.

Evaluating equation (8) for numeric tasks is the same calculation as in the classification task case, albeit at high dimension. There is no shortcut to performing this calculation in general. Nevertheless, in the example to follow, we were able to calculate a contributor's $I(A;J)$ with an answer space of size $s = 400$ in a few seconds using a quick-and-dirty Python script.

Additionally, some special cases allow simplification.

In the case of a uniform answer prior, $p_a = \frac{1}{s}$, equation (8) reduces to

$$\begin{aligned} I(A;J) &= \frac{1}{s}\sum_{a,j} p_{j|a} \log(s \cdot p_{j|a}) \\ &= \log s + \frac{1}{s}\sum_{a,j} p_{j|a} \log p_{j|a}. \end{aligned}$$

This form follows the definition $I(A;J) = H(A) - H(A|J)$, with $H(A) = \log s$ and $H(A|J) = -\frac{1}{s}\sum_{a,j} p_{j|a} \log p_{j|a}$.

If we further assume that we can extend the answer space indefinitely in the positive and negative directions and that $p_{j|a}$ depends only on the difference $\delta \equiv j - a$, writing $p_\delta \equiv p_{j|a}$, we then have

$$H(A|J) = -\sum_\delta p_\delta \log p_\delta.$$

Now that we've reduced the sum to one dimension, if we interpolate p_δ by a continuous probability distribution $f(x)$, we can consider $H(A|J)$ as a Riemann approximation of the *differential entropy* $h[f] \equiv -\int_{\mathbf{R}} dx f(x) \log f(x)$. That is,

$$I(A;J) \approx \log s - h[f].$$

From this it's clear where we might expect the approximation to break down: for values of $h[f]$ close to 0 and close to $\log s$. However, the integral $h[f]$ has a known closed form for many commonly-used continuous probability distributions [18], making the approximation quite versatile.

The case in which $p_{j|a}$ is a Gaussian function of $\delta = j - a$ is of special importance, since a product of Gaussians is also a Gaussian, as is a convolution of Gaussians. This yields closed-form approximations to higher-order quantities like $I(A;J;J')$, as we'll see later.

EXAMPLES

In this section we will present the results of applying our metrics to human judgment tasks from several domains. In addition, we will show how online estimation of contributor

Contributor	$I(A;J)$
1	0.94 bits
2	1.06 bits
3	1.03 bits
4	1.15 bits
5	1.05 bits

Table 2. Mutual information for the contributors in Wilson's experiment.

Patient	Resolution	#2		#3		#4	
2	4	4	3.91	3	2.58	3	1.10
11	4	4	3.91	4	2.32	4	3.32
20	2	1	-2.75	3	-0.08	2	1.16
22	2	2	0.93	2	1.06	2	1.16
32	3	3	2.03	2	0.10	3	2.36
39	3	3	2.03	4	2.58	3	2.36

Table 3. Pointwise mutual information for several patients in Wilson's experiment, for contributors 2, 3, and 4.

$I(A;J'\|J)$	#1	#2	#3	#4	#5
First #1	0.31	0.40	0.37	0.43	0.38
First #2	0.27	0.30	0.33	0.42	0.32
First #3	0.28	0.37	0.33	0.37	0.36
First #4	0.22	0.33	0.25	0.27	0.28
First #5	0.26	0.34	0.33	0.38	0.35

Table 4. Conditional mutual information for second judgments, given the first contributor.

parameters and real-time scheduling of contributors to questions can improve the efficiency and accuracy of some human judgment tasks.

Wilson's Patients

We'll start by applying our metrics to Wilson's data in Dawid and Skene [6], the paper which popularized the use of the expectation-maximization algorithm for hidden-data problems. The data is taken from real pre-operative assessments by 5 contributors of 45 patients' fitness for anesthesia. The judgment and answer spaces are a set of four categories numbered 1, 2, 3, 4.

Each contributor $h \in \{1,2,3,4,5\}$ gives a single judgment $j \in \{1,2,3,4\}$ to each patient $q \in \{1,\ldots,45\}$, except for contributor #1, who gives three (presumed independent) judgments for each patient.

Dawid and Skene proceed as follows. The contributors are modeled using confusion matrices: Each observer is assigned a 4×4 matrix whose j,j' entry is the probability of that observer giving judgment j' when a patient's actual condition is j. Each of the four conditions has a model parameter representing its prior probability for any given patient. Using the expectation-maximization algorithm [7, 12], the 315 judgments collected from the contributors are used to make maximum-likelihood estimates of the model parameters and priors, and these parameter estimates are used to estimate the actual condition of each patient.

We duplicated Dawid and Skene's results, up to what we believe to be an error in their final estimates for patient 7, where we get probabilities of ≈ 0.981 and ≈ 0.019 for categories 1 and 2 respectively, instead of Dawid and Skene's 0.986 and 0.014. We used the maximum-likelihood parameter estimates to form point estimates of the information-theoretic quantities of interest to us.

$I(A;J,J')$	#1	#2	#3	#4	#5
#1	1.25	1.33	1.31	1.36	1.32
#2		1.37	1.40	1.48	1.39
#3			1.36	1.40	1.39
#4				1.42	1.43
#5					1.40

Table 5. Mutual information for two combined contributors.

$I(A;J;J')$	#1	#2	#3	#4	#5
#1	0.63	0.67	0.66	0.72	0.68
#2		0.76	0.70	0.73	0.73
#3			0.69	0.78	0.70
#4				0.87	0.77
#5					0.70

Table 6. Multivariate mutual information for two contributors.

Using the priors and confusion matrices thus obtained (corresponding to Dawid and Skene's Table 2), we measured $I(A;J)$ for each contributor J. The results are shown in table 2.

From the prior $p_a = [0.40, 0.42, 0.11, 0.07]$, the entropy of the answer priors is $H(A) \approx 1.67$ bits.

We also computed pointwise judgment values using the estimated answers (Dawid and Skene's Table 4). In cases where the estimated answer was a probability distribution, we computed the expected judgment value over this distribution.

In table 3, we list the contributors' judgments and the corresponding values of ΔI, in bits, for a selection of patients for each of contributors 2, 3, and 4.

Note that in many cases, a contributor giving the wrong answer still gives us a positive amount of information, because the correct answer is more likely given that judgment. This is to be expected with an answer space of cardinality larger than two: An answer that a contributor is known to confuse with the correct answer narrows down the possibilities for us.

We also computed combined, conditional, and mutual information for pairs of contributors.

Table 4 contains the conditional mutual information between correct answers and a second contributor, conditioned on a first contributor.

Table 5 shows the mutual information between correct answers and two combined contributors.

Table 6 shows multivariate mutual information between correct answers and two contributors.

These data make it clear that although contributor #4 gives us the most information in a single judgment ($I(A;J_4) \approx 1.15$ bits), a second judgment from the same contributor has low expected additional value ($I(A;J_4|J_4) \approx 0.27$ bits). Pairing #4 with #2 gives the most total informa-

tion ($I(A; J_2, J_4) \approx 1.48$ bits), with relatively little overlap ($I(A; J_2; J_4) \approx 0.73$ bits, which is less than any other pairing except $I(A; J_1; J_4) \approx 0.72$ bits).

Welinder and Perona's Bounding Boxes

Next, we considered one of Welinder and Perona's bounding box experiments, introduced previously. In this experiment, 343 contributors gave judgments on 24,642 images, with no image receiving more than 4 contributors' judgments. For simplicity, we restricted our attention to the x-coordinate of the left edge of each bounding box, and we assumed a uniform prior over the columns of each image. Thus the problem is one of estimating an integer value from 0 to $s-1$, where s is the number of columns in the image. s can be different for each image, ranging from a minimum of 88 to a maximum (and mode) of 500.

We chose a simple model for contributor performance, in which each contributor's judgments follow a Gaussian distribution with parameters for bias and variance: $J \sim N(a - \eta, \sigma^2)$, where the *bias* η and *variance* σ^2 are contributor-dependent parameters. The probability of a contributor giving judgment j when the correct answer is a is thus

$$p_{j|a} \propto e^{-\frac{1}{2}\left(\frac{j-a+\eta}{\sigma}\right)^2}.$$

We also assumed an agnostic prior $p_a = \frac{1}{s}$, so that the probabilities of answers given judgments are simply $p_{a|j} \propto p_{j|a}$.

The normalizing constant for the continuous Gaussian distribution is $(\sigma\sqrt{2\pi})^{-1}$, but since we are still assuming a discrete space of possible answers (integer pixel values from 0 to $s-1$), we must calculate the correct factor by brute force, summing

$$\sum_{j=0}^{s-1} e^{-\frac{1}{2}\left(\frac{a-j-\eta}{\sigma}\right)^2}.$$

In particular, the tails of the distribution are cut off by the lateral boundaries of the image.

The continuous approximation to the mutual information $I(A|J)$ is

$$I(A; J) \approx \log s - \log(\sigma\sqrt{2\pi e}) = \log \frac{s}{\sqrt{2\pi e \sigma^2}}. \qquad (12)$$

The continuous approximation breaks down, as expected, for values of σ close to 0 and close to $s/\sqrt{2\pi e}$. We can improve it somewhat simply by truncating it at these values. In figure 2 we plot this corrected approximation along with the actual values of $I(A; J)$ for various values of η and σ^2, with $s = 100$.

The continuous approximation also breaks down when the bias η approaches s in absolute value. This is more a property of our simplistic model than a failure of the approximation, because the model asserts that contributors with large η must be giving uninformative judgments close to one or the other edge of the answer space. The continuous approximation to $I(A; J)$, on the other hand, has no η dependence, because it extends the answer space to the entire real line, on which a contributor with large bias but small variance is as valuable to us as one with small bias and the same variance.

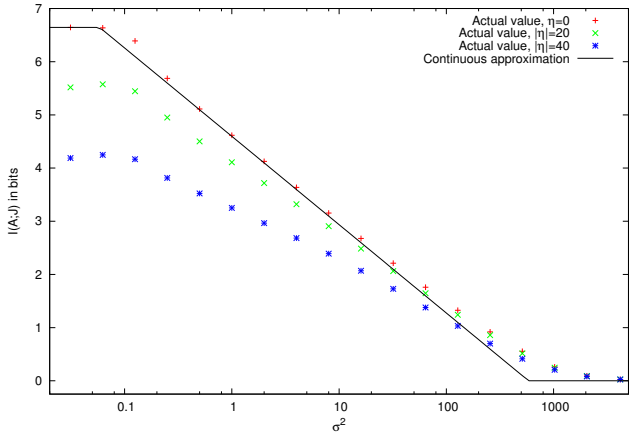

Figure 2. $I(A; J)$ **in the bias-and-variance model as a function of the variance** σ^2**, for three values of the bias** η**, with an answer space of size** $s = 100$**. The solid line is the truncated continuous approximation.**

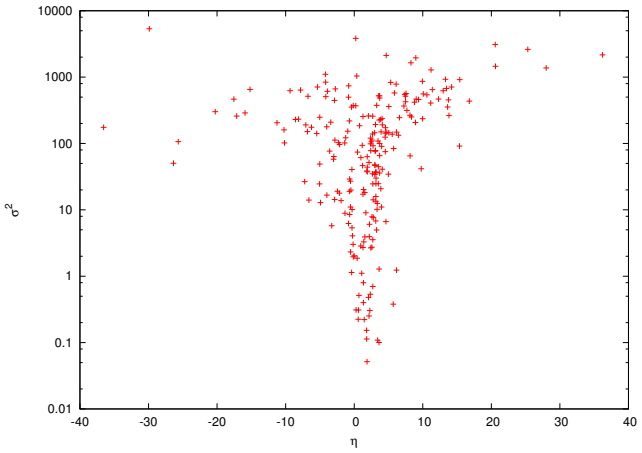

Figure 3. Scatter plot of maximum-likelihood bias and variance for 229 contributors in one of Welinder and Perona's experiments.

The fact that the product of two Gaussian functions is also a Gaussian makes it very easy to estimate correct answers given judgments and model parameters. We implemented the expectation-maximization algorithm over this model and ran it on data from one of Welinder and Perona's experiments.

We note again the important limitation of point parameter estimation approaches such as expectation-maximization, that maximum-likelihood estimators of model parameters are unreliable when there is little data. In particular, the maximum-likelihood estimate $\hat{\sigma}^2$ of the parameter σ^2 for any contributor with only one judgment is zero. (The unbiased estimator $\frac{n}{n-1}\hat{\sigma}^2$ isn't helpful either, as it's undefined for $n = 0$.) A more statistically rigorous approach would be to use marginal integration rather than point parameter estimation, but we chose our approach for the sake of clarity and will simply omit the 114 such contributors in what follows.

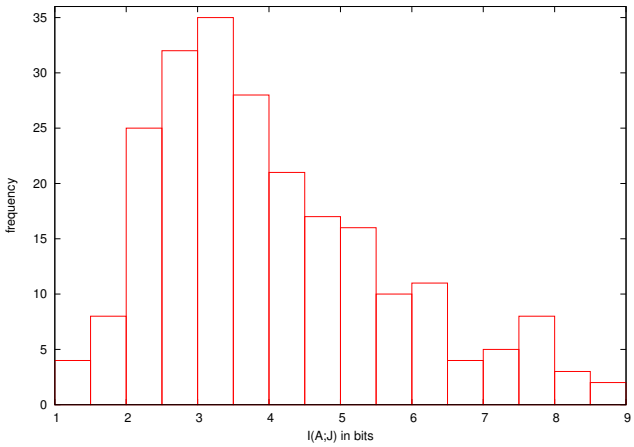

Figure 4. Histogram of mutual information $I(A;J)$ for the contributors in one of Welinder and Perona's experiments, with $s = 500$.

A scatter plot of the maximum-likelihood estimates of contributor parameters η and σ^2 for the remaining 229 contributors is presented in figure 3.

Using these parameter estimates, we calculated the mutual information $I(A;J)$ for each contributor for a hypothetical image of width $s = 500$ (the most common value in the experiment). figure 4 shows a histogram of the results. The upper bound on $I(A;J)$ is $H(A) = \log 500 \approx 9$ bits.

Using the continuous approximation, it's easy to work out higher-order information measures. The posterior probability of answer a given judgments j and j' is

$$p_{a|j,j'} = p_a \cdot p_{j,j'|a} = p_a \cdot p_{j|a} \cdot p_{j'|a}$$

under the assumption of contributor independence. As long as these three factors are all Gaussian, their product is also Gaussian.

For example, if p_a is a constant, $p_{j|a}$ is a Gaussian with variance σ^2, and $p_{j'|a}$ is a Gaussian with variance σ'^2, then $p_{a|j,j'}$ is a Gaussian with variance $\frac{\sigma^2\sigma'^2}{\sigma^2+\sigma'^2}$. That is,

$$H(A|J,J') \approx \log\left(\sigma\sigma'\sqrt{\frac{2\pi e}{\sigma^2+\sigma'^2}}\right).$$

Using the formula $I(A;J,J') = H(A) - H(A|J,J')$ for the combined mutual information, we can compute the information overlap between two contributors with respective variances σ^2 and σ'^2 as

$$\begin{aligned} I(A;J;J') &= I(A;J) + I(A;J') - I(A;J,J') \\ &= H(A) - H(A|J) - H(A|J') + H(A|J,J') \\ &\approx \log\frac{s}{\sqrt{2\pi e(\sigma^2+\sigma'^2)}}. \end{aligned}$$

Comparing this to equation (12), we see that there is little to be gained by pairing a contributor having high variance with one having low variance: The information given to us by the latter is mostly redundant. It's more efficient to trust low-variance contributors and to pair high-variance contributors with each other.

APPLICATION TO SCHEDULING

Scheduling of human resources is an active area of research. Yan et al. [33] discuss optimal assignment of contributors to questions for a binary classification task with a logistic contributor model. Welinder and Perona [30] present an algorithm that adaptively categorizes contributors as "experts" or "bots" and requires fewer judgments per question as it identifies more and more contributors. Additionally, Sheng, Provost, and Ipeirotis [25] discuss several strategies in the context of cost-effectively collecting binary classification labels to be used as training data for an ML classifier.

We'll demonstrate question-contributor scheduling using mutual information conditioned on one or more points, equations (4) and (6). We conducted the following three experiments, one using simulated data and two using real data.

Wilson's Patients

First, we simulated a continuation of Wilson's experiment, with a pool of 50 new patients drawn from the estimated prior distribution and contributors' judgments drawn from their estimated confusion matrices.

We implemented a simple queuing system in which a patient with condition A is drawn at random and assigned to an available contributor, where "available" means "has given the fewest number of judgments so far".

The first contributor J for each patient is chosen randomly from the set of available contributors. Once every patient has been judged by one contributor ($J = j$), we repeat the process to get a second judgment. In control runs, we choose the second contributor randomly again from the pool of available contributors. In test runs, we instead choose for each patient the available contributor J' that maximizes $E_{A,J'|J=j}[\Delta I_{a,j'|j}]$. Using the same contributor both times for one patient is permitted in both cases.

Once two judgments have been collected for each patient, we look at the patient's actual condition a and use the estimated posterior probability of that condition $p_{a|j,j'}$ from the two judgments $J = j$ and $J' = j'$ as a score for our resolution. (This does away with the question of thresholding answer probabilities, which isn't relevant to the experiment.) So a perfect resolution gets a score of 1.0, and a completely incorrect resolution gets a score of 0.0.

We repeated the experiment ten thousand times for each of the control and test runs and computed mean scores of 0.876 for the control group and 0.937 for the test group. That is, using the most likely answer for each patient, if we assign two contributors randomly, we can expect to get the patient's condition correct 87.6 percent of the time, and if we assign the second contributor to maximize expected information gain, we get the patient's condition correct 93.7 percent of the time.

Freebase Reconciliation

For the second experiment, we wished to demonstrate the benefit of a scheduling algorithm even while bootstrapping our estimates of contributor parameters.

Freebase (`http://www.freebase.com`) is a structured knowledge base with an underlying graph model: Topics (graph vertices) are connected to each other by subject-predicate relations (edges).

We considered a Freebase curation task [16]. Sixteen human contributors, engaged in an ongoing basis for this and other tasks, were presented with pairs of Freebase entities representing politicians and political parties and, for each pair, were asked if the politician "is/was a member of" the party. The allowed judgments were "yes", "no", and "skip" (to indicate a problem with a question). Five to seven judgments were collected for each pair, so not every contributor saw every question. There were 10,750 questions and 54,201 judgments in total.

We used a three-by-three confusion matrix model for contributor behavior. We recreated the collection process as follows: First we collected all available judgments for each of 250 questions. We ran the expectation-maximization algorithm over these judgments to obtain initial estimates of the answer priors and contributor confusion matrices.

For the remaining 10,500 questions, we recreated the data collection process using a scheduling algorithm that chose contributors for each question and yielded their real answers from the original data set. This allowed us great flexibility and control in repeating the same experiment under different conditions, without having to collect new data.

In the control groups, the scheduling algorithm chose n contributors for each question, randomly and uniformly without replacement, from the set of contributors from whom a judgment was available for that question. In the test group, the algorithm again chose n contributors, selecting at each step the available contributor with the highest expected information contribution, just as in the above experiment on Wilson's data. The algorithm was free to drop contributors whose judgments were found to be less informative. After every 1,500 questions, we reran the expectation-maximization algorithm over the current set of accumulated judgments, to refine our estimates of the model parameters.

Lacking *a priori* known correct answers to every question, we scored this experiment as follows. For a given set of posteriors from an experimental run, we computed the element-wise dot product of this set with the set of posteriors estimated by running expectation-maximization over the entire data set. We then computed a normalized score by dividing this dot product by the product of the total-data posterior with itself. (We dropped 6 questions whose answer was determined to be "skip".)

For the control groups, we obtained a score of 0.86 with $n = 2$, a score of 0.90 with $n = 3$), a score of 0.90 with $n = 4$, and a score of 0.91 with $n = 5$. For the test groups, we obtained a score of 0.89 with $n = 2$ and a score of 0.91 with $n = 3$.

In other words, by choosing contributors with a high expected information contribution, we were able to achieve results with only two or three contributors per question comparable to those obtained from four or five contributors per question chosen arbitrarily.

Galaxy Zoo 2

The Galaxy Zoo project [19] is an ongoing crowdsourcing effort to classify galaxies from deep-field surveys. The component known as Galaxy Zoo 2 [32] (`http://zoo2.galaxyzoo.org`) asked its users to answer a sequence of questions in order to identify each object. For example, if a user identified an object as an elliptical galaxy, the next question asked the galaxy's shape; on the other hand, if the user identified the object as a spiral galaxy, subsequent questions would ask for the number and packing density of the spirals.

To simplify the data set, we first collapsed each set of judgments for a given object and contributor into a single classification judgment. We dropped several questions in order to reduce the dimension of the answer space to the following nine classes: three classes of elliptical galaxies (round, cigar-shaped, and intermediate), one class for disk galaxies seen edge-on, four classes for disk galaxies with or without bars and with or without spirals, and one class for non-galaxies. This simplification made judgment resolution easily tractable using a nine-by-nine confusion matrix for each contributor.

As in the Freebase experiment above, we ran the experiment by selectively replaying contributors' actual judgments. In order to have sufficient flexibility in scheduling, we needed a data set comprising contributors with lots of judgments. From the 13,880,502 recorded judgments of 355,990 objects by 83,906 contributors, we iteratively subtracted objects with fewer than 15 judgments each and contributors with fewer than 800 judgments each, repeating the process until the set stabilized at 6,159,423 judgments of 272,270 objects by 2,956 contributors.

We generated reference classifications for each object by running expectation-maximization over all the judgments in this data set using the confusion matrix model and taking the most likely classification in each case. We computed the per-question entropy to be 2.72 bits. Contributor mutual information ranged from 0 bits (for a contributor who gave 997 judgments, all "non-galaxy") to 1.68 bits per question.

We scored our algorithm and two control-group algorithms at various points by running expectation-maximization over all the data they had collected so far and computing the proportion of the resulting classifications that matched the reference classifications.

Each algorithm was given ten rounds in which to assign objects to contributors. Each round consisted of collecting 92 judgments from each contributor.[4]

[4]This batch size was chosen as a computational convenience, with the mean number of judgments collected per round approximately equal to the total number of objects, although the judgments were not evenly distributed between the objects. Also, some contributors ran out of judgments before the end of the experiment.

Our algorithm, in order to play fair, started with no knowledge of the contributors' confusion matrices. In the first round, we had it assign to each contributor the set of 92 available objects (where "available" means "classified by this contributor in the real experiment but not yet in this simulation") with the most judgments already collected, thus collecting judgments for only a small number of objects. Although this approach is inefficient at determining actual classifications, it has the benefit of giving the algorithm a good initial estimate of the contributors' confusion matrices.[5] In each subsequent round, the algorithm formed new maximum-likelihood estimates of the confusion matrices by running expectation-maximization over all judgments collected so far, and then assigned to each contributor the subset of 92 available objects with the highest values of mutual information conditioned on the judgments already collected for that object, as defined by equation (6).

We compared our algorithm to a "least-answered" algorithm, which assigned to each contributor at each round the subset of 92 available objects with the fewest judgments collected so far, thus attempting to equalize the number of judgments collected across all objects.

We also included a "random-assignment" algorithm, which assigned to each contributor at each round a randomly chosen subset of 92 available objects.

We scored each of the three algorithms after each round of judgment collection. The results are presented in figure 5. Our algorithm showed poor results after the first round in which it accepted redundant judgments in order to bootstrap its estimates of the confusion matrices, but then quickly caught up and consistently outperformed the two control algorithms. In later rounds the gap closed because there were fewer available judgments to choose from, a limitation of the "replay" nature of this experiment, so the three algorithms' judgment sets and classifications began to converge to the full data set and its reference classifications.

We expect that we have only scratched the surface of the scheduling problem. Although the models we discussed above for search and numeric tasks are too one-dimensional to benefit from online contributor scheduling, one can imagine more sophisticated statistical models that incorporate a notion of domain expertise, for which a scheduling algorithm like those above would progressively identify the domain of a question and assign contributors skilled in that domain.

CONCLUSIONS

We developed the pointwise mutual information metric $\Delta I_{a,j}$, a measure of the amount of information in a contributor's judgment that we use to estimate the correct answer to the question, together with its expected value, the mutual information metric $I(A;J)$, which measures the average information obtained per question from a contributor.

[5]When we tried the experiment without this seeding approach, the results were no better than the control groups, we believe because of the sparsity of the incidence matrix between contributors and objects in such a large-scale crowdsourcing project.

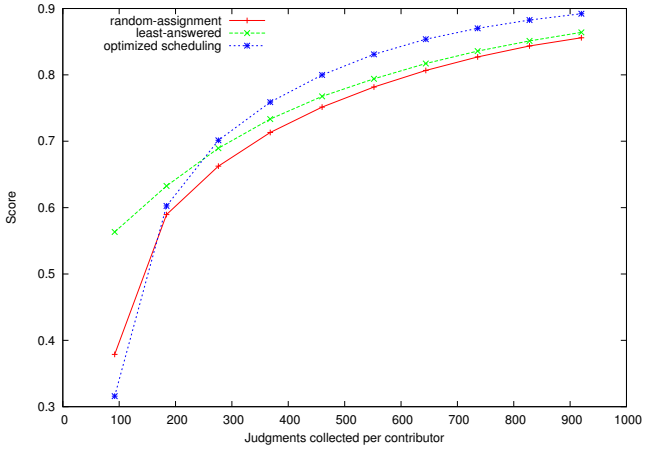

Figure 5. Scoring of three judgment collection algorithms for the Galaxy Zoo 2 experiment at each round of data collection. Our algorithm performs poorly at first because it focuses on getting good initial estimates of confusion matrices, then quickly overtakes the two control group algorithms.

We then explored the use of conditional and multivariate mutual information to capture the interaction between judgments from different contributors (or between multiple judgments from the same contributor).

We showed how to compute these quantities in an experimental environment, and how they can be adapted to different statistical models of contributor behavior, specifically discussing the differences between confusion matrix models and probability-correct models. We provided a simplification that makes our metrics easy to compute even over a very large answer space.

We then described how to use them to optimize question-contributor scheduling, demonstrating the utility of this technique in three experiments.

We believe there is much potential in the information-theoretic approach to contributor evaluation. We will conclude by briefly introducing several possible extensions to the work presented in this paper.

Evaluating Resolution Algorithms

It's important to emphasize that mutual information does not measure correctness of the judgments given. Judgments are collected in order to help us find the correct answers to questions, but we acknowledge that the end result of the collection process must inevitably be *estimates* of the correct answers, not the correct answers themselves. We are only measuring the amount by which contributors help us to arrive at these estimates.

Therefore, our metrics depend very strongly on the estimation techniques used. For example, if the answer to each question is estimated using majority voting among the judgments collected for that question, than we are rewarding conformity and nothing else. If we use a probability-correct model where a confusion matrix model would be more appropriate, then we are ignoring (and punishing) the judgments of biased workers. If we use a probability-correct model for questions

over interval scales, then we reward the single contributor the model decides is most likely correct and punish the others even if their answers are very close to the estimated correct answer.

In summary, the value of contributors' judgments is linked to our use of those judgments. If we make good estimates of the answers to our questions, then we value good judgments highly, and if we make poor estimates, then we place little value on good judgments.

Turning this problem around, we arrive at another application for our metrics: as a test of the health of a resolution algorithm. If an algorithm yields low values of $I(A;J)$ for many contributors, then by definition, that algorithm is making very little use of those contributors' judgments. Another algorithm that yields higher values of $I(A;J)$ is making more use of the same judgments and is likely to be producing better-informed estimates of the correct answers.

Interaction with Reproducibility

The use of Krippendorff's alpha as a measure of the overall statistical health of the collection process can be coupled with the information-theoretic approach.

Since alpha is computed using only the judgments and does not depend on the resolution algorithm, it can be used as an early warning system for ill-defined tasks or poorly-briefed contributors.

We expect that, in addition to Paritosh's argument [20] that reproducibility is a prerequisite for accurate resolutions, reproducibility is also a prerequisite for obtaining high value from judgments.

The reason for this is that a task with low alpha is one in which judgments for the same question are not well-correlated with each other, and therefore cannot be well correlated with the correct answers. Hence, a set of judgments with low alpha must necessarily have low information, under any resolution mechanism.

Establishing a mathematical or empirical relationship between alpha and mutual information is a potential topic for future work.

Normalized Information

$I(A;J)$ measures contributor performance in extensive physical units. In some applications, we may instead be more interested in a normalized (intensive) score for contributor performance, on a fixed scale.[6]

We can exploit the inequality

$$0 \leq I(A;J) \leq H(A),$$

which expresses the limits that a worst-case contributor yields no information to the resolution process and that a perfect contributor gives us exactly as much information as is present in the question itself.

[6]For the non-physicists: *extensive* quantities are those which depend on the amount of stuff being measured and are additive, like mass; and *intensive* quantities are those which are invariant under the quantity of stuff being measured, like density.

Thus, we normalize $I(A;J)$ by dividing by the theoretical optimal performance $H(A)$, defining a normalized score

$$C(A;J) \equiv \frac{I(A;J)}{H(A)}$$

for a given contributor, with $0 \leq C(A;J) \leq 1$.

Weighted Entropy

If we place a higher value on certain outcomes and we wish to incentivize contributors accordingly, we can attach weights $\boldsymbol{w}$ to the outcomes in equation (1), subscripted either by a or by a and j together, yielding weighted information $w_a \cdot \Delta I_{a,j}$ or $w_{a,j} \cdot \Delta I_{a,j}$. Then equation (2) becomes

$$I_{\boldsymbol{w}}(A;J) = \sum_a w_a \sum_j p_{a,j} \log \frac{p_{a|j}}{p_a}$$

or

$$I_{\boldsymbol{w}}(A;J) = \sum_{a,j} w_{a,j} p_{a,j} \log \frac{p_{a|j}}{p_a},$$

respectively.

For instance, if we consider it especially important to identify objects of class a_2 correctly in the example above, we can use weights $w_1 = 2$ and $w_2 = 1$, so that our contributor's work on objects the resolution algorithm determines to be in class a_2 is given twice as much weight as that contributor's work on objects of class a_1.

On the other hand, if we wish specifically to discourage mistaking objects of class a_2 for class a_1 (as our contributor is known to do), we can set $w_{2,1} = 2$ and $w_{a,j} = 1$ otherwise, thus punishing our contributor for making that mistake. This works as long as $\Delta I_{2,1} < 0$; if $\Delta I_{2,1} > 0$, then the mistake actually helps rather than hinders the resolution process, so the added weight incentivizes the mistake.

ACKNOWLEDGMENTS

Many thanks to colleagues Praveen Paritosh, Ronen Vaisenberg, Reilly Rose Hayes, Jamie Taylor, Stefano Mazzocchi, Ellen Spertus, Colin Evans, Viral Shah, Micah Saul, Robert Klapper, and Dario Amodei, for valuable comments, conversations, and suggestions during this work.

We are especially grateful to Chris Lintott and Arfon Smith for providing us with data from Galaxy Zoo 2, to Peter Welinder for sharing with us the raw data from an early bounding-box experiment, and to Michael Shwe for providing us with the data from the Freebase political parties task.

REFERENCES

1. Beal, M. J. *Variational algorithms for approximate Bayesian inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.
2. Card, S. K., Moran, T. P., and Newell, A. The model human processor: An engineering model of human performance. In *Handbook of Perception and Human Performance*, K. R. Boff, L. Kaufman, and J. P. Thomas, Eds., vol. 2: Cognitive Processes and Performance. Wiley-Interscience, 1986, ch. 45, 1–35.

3. Card, S. W., and Mohan, C. K. Ensemble selection for evolutionary learning using information theory and price's theorem. In *Proc. GECCO 2006*, ACM Press (2006), 1587–1588.
4. Chen, Y., Dimitrov, S., Sami, R., Reeves, D. M., Pennock, D. M., Hanson, R. D., Fortnow, L., and Gonen, R. Gaming prediction markets: Equilibrium strategies with a market maker. *Algorithmica 58*, 4 (2010), 930–969.
5. Chi, E. H., and Mytkowicz, T. Understanding the efficiency of social tagging systems using information theory. In *Proc. HT 2008*, ACM Press (2008), 81–88.
6. Dawid, A. P., and Skene, A. M. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* (1979), 20–28.
7. Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39*, 1 (1977), 1–38.
8. Dow, S., Kulkarni, A., Bunge, B., Nguyen, T., Klemmer, S., and Hartmann, B. Shepherding the crowd: managing and providing feedback to crowd workers. In *Ext. Abstracts CHI 2011*, ACM Press (2011), 1669–1674.
9. Gelfand, A. E., and Smith, A. F. M. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association 85*, 410 (June 1990), 398–409.
10. Geman, S., and Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 (1984), 721–741.
11. Hanson, R. Logarithmic market scoring rules for modular combinatorial information aggregation. *The Journal of Prediction Markets 1*, 1 (2007), 3–15.
12. Hartley, H. O. Maximum likelihood estimation from incomplete data. *Biometrics 14*, 2 (June 1958), 174–194.
13. Hastings, W. K. Monte carlo sampling methods using markov chains and their applications. *Biometrika 57*, 1 (1970), 97–109.
14. Hick, W. E. On the rate of gain of information. *Quarterly Journal of Experimental Psychology 4* (1952), 11–26.
15. Ipeirotis, P. G., Provost, F., and Wang, J. Quality management on Amazon Mechanical Turk. In *Proc. HCOMP 2010*, ACM Press (2010), 64–67.
16. Kochhar, S., Mazzocchi, S., and Paritosh, P. The anatomy of a large-scale human computation engine. In *Proc. HCOMP 2010*, ACM Press (2010), 10–17.
17. Krippendorff, K. Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement 30*, 1 (1970), 61–70.
18. Lazo, A. V., and Rathie, P. On the entropy of continuous probability distributions. *IEEE Transactions on Information Theory 24*, 1 (1978), 120–122.
19. Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., Murray, P., and van den Berg, J. Galaxy Zoo: Morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society 389*, 3 (2008), 1179–1189.
20. Paritosh, P. Human computation must be reproducible. In *Proc. First International Workshop on Crowdsourcing Web Search* (2012), 20–25.
21. Quinn, A. J., and Bederson, B. B. Human computation: a survey and taxonomy of a growing field. In *Proc. CHI 2011*, ACM Press (2011), 1403–1412.
22. Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal 27*, 3 (1948), 379–423. Part 1 of 2.
23. Shannon, C. E., and Weaver, W. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
24. Shaw, A. D., Horton, J. J., and Chen, D. L. Designing incentives for inexpert human raters. In *Proc. CSCW 2011*, ACM Press (2011), 275–284.
25. Sheng, V. S., Provost, F., and Ipeirotis, P. G. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proc. KDD 2008*, ACM Press (2008), 614–622.
26. Simpson, E., Roberts, S., Psorakis, I., and Smith, A. Dynamic Bayesian combination of multiple imperfect classifiers. *ArXiv e-prints* (June 2012).
27. Tanner, M. A., and Wong, W. H. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association 82*, 398 (June 1987), 528–540.
28. Walsh, B. Markov chain Monte Carlo and Gibbs sampling, 2004. Lecture Notes for EEB 581.
29. Welinder, P., Branson, S., Belongie, S., and Perona, P. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, vol. 23, NIPS (2010), 2424–2432.
30. Welinder, P., and Perona, P. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Proc. CVPR 2010*, IEEE (2010), 25–32.
31. Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., and Movellan, J. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems 22* (2009), 2035–2043.
32. Willett, K. W., Lintott, C. J., et al. In preparation.
33. Yan, Y., Rosales, R., Fung, G., and Dy, J. Active learning from crowds. In *Proc. ICML 2011*, IMLS (2011), 1161–1168.
34. Yang-Peláez, J., and Flowers, W. C. Information content measures of visual displays. In *Proc. InfoVis 2000*, IEEE (2000), 99–103.