

# An Information-Theoretic Metric for Collective Human Judgment

**Tamsyn Waterhouse**

Google, Inc.

345 Spear Street, 4th Floor  
San Francisco, CA 94105-1689

## 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.

## Introduction

### Background

Human computation, as defined by Quinn and Beder-son (2011) (which see 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 work under a 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 (2010). Each question is presented independently to a number of contributors, and their judgments are synthesized to form an estimate of the correct answer.

Collective human computation 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 or no supervision. There is a strong 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.

In both cases, there is a growing need for objective measurements of worker performance.

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. (2011). Shaw, Horton, and Chen (2011) 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 (2010), 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 contribu-

tors' 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 (2010) 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.

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.

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.

## Notation

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.

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.

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  $\Delta I$  can be negative,  $I(A; J)$  is always nonnegative: 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.

## Example

Say we have a task involving classifying objects  $q$  into 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.

We have the following table of outcomes:

| $a$   | $j$   | $p_{a,j}$ | $p_{a j}$     | $I(A = a J = j)$ | $\Delta I_{a,j}$ |
|-------|-------|-----------|---------------|------------------|------------------|
| $a_1$ | $a_1$ | 0.5       | $\frac{1}{2}$ | 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            |

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.

## 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  $H(A|J, \dots)$ . 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*

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

Below, we'll consider two situations: assigning a second contributor  $J'$  to a question after receiving a judgment from a first contributor  $J$  (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.

## Sequentially

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

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

which is similar in form to 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 3 over all three random variables rather than just  $A$  and  $J'$ . The result is the *conditional mutual information*

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

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  $J'$ . 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. It might not be fair to reward a contributor's judgments based on their statistical interaction with another contributor's judgments, but we can use this as a measure of inefficiency in our system. We receive an amount of information equal to  $I(A; J) + I(A; J')$ , 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.

## More than Two Judgments

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

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

which gives the total information we get from  $k$  judgments, and the conditional mutual information

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

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

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

for measuring information overlap.

## Practical Calculation

### Equations

In practice, we generally have a 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.

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

Here is how to compute our mutual information quantities using the typically-known probabilities. Pointwise mutual information, 1, is

$$\Delta I_{a,j} = \log \frac{p_{a|j}}{p_a} = \log \frac{p_{j|a}}{\sum_{a'} p_{a'} p_{j|a'}}. \quad (6)$$

Mutual information, 2, is

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

Pointwise conditional mutual information, 3, is

$$\Delta I_{a,j'|j} = \log \frac{p_{a|j,j'}}{p_{a|j}} = \log \left( p_{j'|a} \frac{\sum_{a'} p_{a'} p_{j|a'}}{\sum_{a'} p_{a'} p_{j|a'} p_{j'|a'}} \right). \quad (8)$$

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

$$\begin{aligned} E_{A,J'|J=j}[\Delta I_{a,j'|j}] &= \sum_{a,j'} p_{a,j'|j} \Delta I_{a,j'|j} \\ &= \sum_{a,j'} \frac{p_a p_{j|a} p_{j'|a}}{\sum_{a'} p_{a'} p_{j|a'}} \log \left( p_{j'|a} \frac{\sum_{a'} p_{a'} p_{j|a'}}{\sum_{a'} p_{a'} p_{j|a'} p_{j'|a'}} \right). \end{aligned} \quad (9)$$

Conditional mutual information, 5, is

$$\begin{aligned} I(A; J'|J) &= \sum_{a,j,j'} p_{a,j,j'} \Delta I_{a,j'|j} \\ &= \sum_{a,j,j'} p_a p_{j|a} p_{j'|a} \log \left( p_{j'|a} \frac{\sum_{a'} p_{a'} p_{j|a'}}{\sum_{a'} p_{a'} p_{j|a'} p_{j'|a'}} \right). \end{aligned} \quad (10)$$

Combined mutual information is then

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

and multivariate mutual information is

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

### Computational Complexity

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 examples to follow, 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 (2004) for an introduction).

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 7 has runtime  $O(m \cdot s \cdot t)$ , and computing a second-order quantity such as  $I(A; J, J')$  for all contributor pairs using 10 has runtime  $O(m^2 \cdot s \cdot t^2)$ . Higher-order quantities become challenging, though: In general, computing  $I(A; J_1, \dots, J_k)$  for all contributor  $k$ -sets has runtime  $O(m^k \cdot s \cdot t^k)$ .

By comparison, estimating answers to all questions in a task from a single set of model parameter values can be accomplished in runtime proportional to the total number of judgments received. This number varies depending on the experimental setup and the desired accuracy of the results. In the literature it is typically of order 10 times the number of questions.

### Search Problems

The above discussion works well for classification problems, those in which the same small answer space is shared by all questions, and we have enough data to estimate priors  $p_a$  and confusion matrix elements  $\pi_{a,j}$ .

A good example of this sort of problem is Galaxy Zoo (Lintott et al. 2008), which uses volunteers to classify galaxy morphology from photographs. Contributor behavior for classification problems can usually be modeled well with confusion matrices  $p_{j|a} = \pi_{a,j}$ , and the small size of the answer space relative to the number of questions allows us to make good estimates of the class priors  $p_a$ .

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, 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 runtime 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  $\pi$  that gives the probability of a contributor answering a question correctly.  $\pi$  may depend on the contributor, the question, or both; see Whitehill et al. (2009) 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  $\pi$ , 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 6 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  $s \rightarrow \infty$ . In 1, we plot  $\frac{I(A; J)}{H(A)}$  for various values of  $s$ .

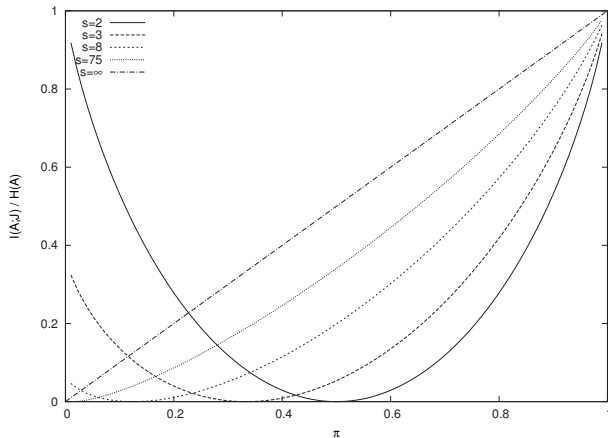


Figure 1:  $\frac{I(A; J)}{H(A)}$  as a function of  $\pi$  and  $s$ .

We can also compute higher-order quantities. 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  $\pi$  per contributor, contributor skill is literally one-dimensional.

In any event, what matters is that for search problems, we can eliminate the sums in 6–10 to produce computationally simple quantities.

## Examples

We'll apply our metrics to Wilson's data in Dawid and Skene (1979). The data is taken from real pre-operative assessments by 5 observers of 45 patients' fitness for anesthesia. The judgment and answer spaces are a set of four categories numbered 1, 2, 3, 4. We duplicated Dawid and Skene's results, using as they did the EM algorithm and a confusion matrix model of observer behavior.

### One Contributor at a Time

Using the priors and confusion matrices thus obtained (corresponding to Dawid and Skene's Table 2), we measured  $I(A; J)$  for each observer  $J$  as follows:

| Observer          | 1    | 2    | 3    | 4    | 5    |
|-------------------|------|------|------|------|------|
| $I(A; J)$ in bits | 0.94 | 1.06 | 1.03 | 1.15 | 1.05 |

Table 1:  $I(A; J)$  for Wilson's observers.

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.

Below are the observers' judgments and the corresponding values of  $\Delta I$ , in bits, for a selection of patients for each of observers 2, 3, and 4.

| Patient | Answer | #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 |
| 32      | 3      | 3 2.03  | 2 0.10  | 3 2.36 |
| 39      | 3      | 3 2.03  | 4 2.58  | 3 2.36 |

Note that in many cases, an observer 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.

### Multiple Contributors

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

Although observer #4 gives us the most information in a single judgment ( $I(A; J_4) \approx 1.15$  bits in 1), a second judgment from the same observer has low expected additional

| $I(A; J' J)$ | #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 2: Mutual information between correct answers and a second observer, conditioned on a first observer.

| $I(A; J, J')$ | #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 3: Mutual information between correct answers and two combined observers.

| $I(A; J, J')$ | #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 4: Multivariate mutual information between correct answers and two observers.

value ( $I(A; J_4|J_4) \approx 0.27$  bits in 2). Pairing #4 with #2 gives the most total information ( $I(A; J_2, J_4) \approx 1.48$  bits in 3), with relatively little overlap ( $I(A; J_2; J_4) \approx 0.73$  bits in 4).

## Scheduling

Now we’ll demonstrate question-contributor scheduling using mutual information conditioned on one point, 9. Scheduling of human resources is an active area of research; see Yan et al. (2011) for a sample.

We simulated a continuation of Wilson’s experiment, with a pool of 50 new patients drawn from the estimated prior distribution and observers’ 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 observer, where “available” means “has given the fewest number of judgments so far”.

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

We repeated the experiment 10000 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 observers randomly, we can expect to get the patient’s condition correct 87.6 percent of the time, and if we assign the second observer to maximize expected information gain, we get the patient’s condition correct 93.7 percent of the time.

## Acknowledgments

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

## References

- Dawid, A., and Skene, A. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* 20–28.
- Dow, S.; Kulkarni, A.; Bunge, B.; Nguyen, T.; Klemmer, S.; and Hartmann, B. 2011. Shepherding the crowd: managing and providing feedback to crowd workers. In *Ext. Abstracts CHI 2011*, 1669–1674. ACM Press.
- Ipeirotis, P.; Provost, F.; and Wang, J. 2010. Quality management on Amazon Mechanical Turk. In *Proc. HCOMP 2010*, 64–67. ACM Press.
- Kochhar, S.; Mazzocchi, S.; and Paritosh, P. 2010. The anatomy of a large-scale human computation engine. In *Proc. HCOMP 2010*, 10–17. ACM Press.
- Lintott, C.; Schawinski, K.; Slosar, A.; Land, K.; Bamford, S.; Thomas, D.; Raddick, M.; Nichol, R.; Szalay, A.; Andreescu, D.; Murray, P.; and van den Berg, J. 2008. Galaxy Zoo: Morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society* 389(3):1179–1189.
- Quinn, A., and Bederson, B. 2011. Human computation: a survey and taxonomy of a growing field. In *Proc. CHI 2011*, 1403–1412. ACM Press.
- Shaw, A.; Horton, J.; and Chen, D. 2011. Designing incentives for inexpert human raters. In *Proc. CSCW 2011*, 275–284. ACM Press.
- Walsh, B. 2004. Markov chain Monte Carlo and Gibbs sampling. Lecture Notes for EEB 581.
- Welinder, P., and Perona, P. 2010. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Proc. CVPR 2010*, 25–32. IEEE.
- Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems* 22:2035–2043.
- Yan, Y.; Rosales, R.; Fung, G.; and Dy, J. 2011. Active learning from crowds. In *Proc. ICML 2011*, 1161–1168. IMLS.