Quantifying information content of recommendations

Tim Muller

University of Luxembourg

Trust Opinions

Agents interact over the internet. Some of these interactions involve an agent without a guarantee that the other agent will perform his task adequately. There are many reasons for an agent not to perform his task adequately, such as monetary gain (e.g. not delivering a purchased good), cutting corners (e.g. delivering nonsense rather than data in a cloud), ineptitude (e.g. an online service being off line), etcetera. Rather than focussing on underlying motives or qualities, we lump these problems together as unsuccessful interactions, or failures. Hence we may observe that some agents are inherently more likely to fail in interactions than others. Furthermore, past interactions with an agent is a predictor of the likelihood of failure in the next interaction.

A model that has precisely these assumptions is the Beta model. In the Beta model, trust opinions are represented as beta distributions. A beta distribution takes two parameters that depend on the number of observed successes and failures. The random variable R in the distribution, has the property that if $R = x$, then the probability of success is x, so $P(E = \mathrm{s}|R = x) = x$.

The Beta model does not support recommendations. In a recent technical report[1], we have formally introduced recommendations of the shape $\mathbb{N} \times \mathbb{N}$. A recommendation (s, f) by agent B (about agent C) is a claim of B that B has interacted $s + f$ times with C, and that s of those interactions were successes, and f failures. An agent A may receive such a recommendation from B, and establish an opinion on the basis of that recommendation. If A knows that B is always honest, he can take the beta distribution based on (s, f). However, in general, B is not always honest.

The shape of the resulting trust opinion of A is not generally a beta distribution. In fact, the resulting trust opinion of A can be characterized as $p \cdot \beta(s + 1, f + 1) + (1 - p) \cdot \sum_{0 \leq i, 0 \leq j} w_{i,j} \beta(i + 1, j + 1)$. The factor p is the probability that B provides an honest recommendation, and the factors $w_{i,j}$ depend both on the probability that B says (s, f) when B actually observed (i, j) and on the probability that B has observed (i, j). In other words, the trust opinion of A based on the recommendation of B is a weighted sum of beta distributions.

Information of Recommendations

The trust opinion based on $(0, 0)$ is the uniform distribution on $[0, 1]$. The trust opinion based on (N, N), for large N, is close to the point distribution on 0.5. Both have an expected value of 0.5, but the former distribution feels somehow less informative than the latter. For one, the (Shannon) entropy of the former is much larger than the entropy of the latter. Another explanation is that the former is based on 0 interactions, whereas the latter is based on $2N$ interactions. We could pick either as a representation for the amount of information of the distribution, but the result of the choice differs. For example, a trust opinion based on $(5, 0)$ has less entropy (more information) than a distribution based on $(3, 3)$, but less interactions. In this section we compare the two types measures.

Interactions

The most straightforward measure of information is the number of interactions required for the distribution. In the case that the trust opinion is a beta distribution, the number of interactions required is obvious. It is less obvious if the trust opinion is a sum of different beta distributions.

We need a measure that provides the right answers for beta distributions, that provides a unique answer for a distribution, and that provides intuitive answers for values between two beta distributions. Say that we have two trust opinions both being the weighted sum of a beta distribution with many interactions and a beta distribution with little interactions; the trust opinion that values the beta distribution with many interactions more is expected to be more informative. The uniqueness requirement is obvious, but not trivial. Considering that $\beta(1, 1) = 0.5 \cdot \beta(2, 1) + 0.5 \cdot \beta(1, 2)$, the measure must be equal for both sides.

We can construct a measure based on the notion that for every s, f, there are a, b such that $\beta(s+1, f+1) = a \cdot \beta(s + 2, f + 1) + b \cdot \beta(s + 1, f + 2)$. Using that notion, it is possible to convert every weighted sum of beta distributions $\sum_i w_i \cdot \beta(s_i + 1, f_i + 1)$, with $s_i + f_i \leq n$, into a sum $\sum_j v_j \cdot \beta(s'_j + 1, f'_j + 1)$ with $s'_j + f'_j = n$. For example, $0.5 \cdot \beta(1, 1) + 0.5 \cdot \beta(2, 1) = 0.75 \cdot \beta(2, 1) + 0.25 \cdot \beta(1, 2)$. We can use the total difference $\sum_j |\frac{1}{n+1} - v_j|$ as a basis for the measure, as it has the property that it is invariant over increases of n. More precisely, we need to multiply that with $\frac{n+1}{2}$, to normalize it to number of experiments, making $\frac{n+1}{2} \cdot \sum_j |\frac{1}{n+1} - v_j|$. Although this measure adheres to the three properties we required, it does have some counterintuitive properties.

[1] http://satoss.uni.lu/members/tim/papers/TrustChainingTechReport.pdf

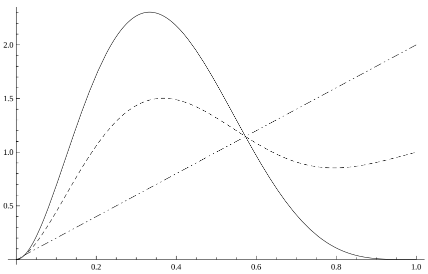

Fig. 1. Solid: Graph based on (2,4): $\beta(3,5)$. Dashed: Graph based on (0,1): $\beta(1,2)$. Dash-dotted: Graph based on weighted sum of the others: $0.5 \cdot \beta(3,5) + 0.5 \cdot \beta(1,2)$

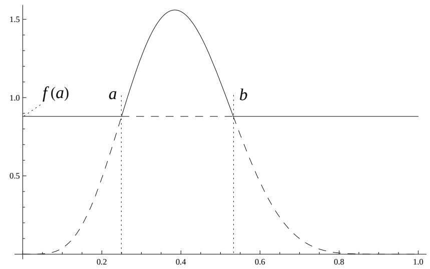

Fig. 2. A beta distribution $\beta(6,9)$ multiplied with 0.5. The line $f(a)$ has the property that $\max(f(a), \beta(6,9))$ has a surface area of 1 in $[0,1]$.

Furthermore, it can only be applied to finite weighted sums of beta distributions.

There is a better measure that is congruent to measuring the number of experiments, based on entropy measures.

Entropy

A standard way of representing information content is entropy. Contrary to the method based on interactions, it generalizes neatly to other distributions. If we have a discrete random variable X, the entropy is given $\sum_x P(X = x) \cdot \log(P(X = x))$. If we have a continuous random variable X, the entropy is given $\int_X f_X(x) \cdot \log(f_X(x))\mathrm{d}x$. Say we want to use the notion of entropy to represent the information content of a trust opinion, then we need to select a random variable. There are two obvious candidates, the (continuous) random variable representing the reliability of an agent (i.e. the random variable that is distributed), or the (discrete) random variable representing the next outcome.

If we are interested in information regarding the reliability of agent C, the natural choice is to look at the entropy of the random variable representing that; R_C. However, we may not actually be interested in the exact value of R_C. Whether R_C is equal to 0.501 or 0.502 may not be important, yet information-wise these two values are treated as completely different values. We can imagine that whether R_C equals 0.01 or 0.02 does matter (and contrary to values near 0.5). But ironically, if the distribution is based on $(0, 10)$ we have more information about R_C then when the distribution is based on $(5, 5)$ (see Figure 1). In other words, if R_C is close to 0 (or 1), the information about R_C increases, but the required precision does too.

An interesting scenario arises, if we pick R_C to be the measure of information. If a recommender is forced to tell the truth with probability p, and can lie otherwise, the information content of his recommendation is that of $p \cdot T + (1 - p) \cdot L$. In Figure 2, we can see an example, where $p = 0.5$ and the truth T is $\beta(6, 9)$, and the lying strategy L is chosen to maximize the total entropy of the graph. For us, an open question is whether we can generally find exact values for a and b, which is required for finding L.

Arguably, we are not interested in the (exact) value of R_C. If we are interested in information regarding the next outcome of C, we should study the entropy of the random variable representing the next outcome; E_C. Recall that $P(E_C = \mathrm{s}|R_C = 0.5) = 0.5$, meaning that the outcome is equivalent to a fair coin flip, which has 1 bit entropy. However, R_C is not a fixed number, but has a distribution, hence we should take the expectation of the entropy of E_C: $\int_0^1 f_{R_C}(x) \cdot H(E_C|R_C = x)\mathrm{d}x$. If R_C happens to be somewhere around 0.5, then increasing the number of interactions is increasing the entropy too (decreasing the information). This measure of entropy on the the distribution based on $(0, 0)$ gives 0.721348 bits of entropy, but $(100, 100)$ gives 0.996402 bits of entropy. We lose $0.996402 - 0.721348 = 0.275054$ bits of information (about the next outcome), when we have more interactions.

The problem with studying the expected entropy of E_C, is that it is strongly biased around 0.5. Let E_C^x and E_C^y be random variables corresponding to E_C given $R_C = x$ and $R_C = y$, respectively. Then we can study the expected relative entropy of E_C^x with E_C^y: $\int_0^1 \int_0^1 f_{R_C}(x) \cdot f_{R_C}(y) \cdot D_K L(E_C^x||E_C^y)\mathrm{d}x\mathrm{d}y$. Essentially, this measures the entropy of the outcome of the interaction with a randomly (according to the distribution f_{R_C}) selected reliability x, relative to the outcome of an interaction with the true machine with unknown reliability y (distributed with f_{R_C}). The important characteristic of this approach is that it is insensitive to whether a number of interactions provide the same outcomes (e.g. many success, little failures), or similar amounts of each outcome, meaning that the information in $(5, 5)$ is identical to the information in $(10, 0)$. Moreover, the entropy of the beta distribution based on s successes and f failures equals $\frac{1}{s+f+2}$. Recall that entropy is the converse of information, so if $s + f$ increases, we expect the entropy to decrease towards 0. This entropy measure combines the best of both worlds, but has the peculiar side effect that the uniform distribution does not provide the minimal entropy (but $\beta(0, 0)$ does).