Crowdtrust - Trustworthy Information From The Crowd Group Project

Giovanni Charles Adam Fiksen Ryan Jackson Sahil Jain John Walker

Emil Lupu

January 5, 2013

Abstract

Outsourcing is becomming increasingly popular in a lot of areas and as a result providing these required services has become big business. Companies are using the Internet to accept jobs from requestors which often require some human intelligence such as tagging images, they then distribute these jobs to a crowd of workers and return the results to the requestor. this process is called crowdsourcing.

At the moment Amazons *Mechanical Turk* is the only real crowdsourcing 'giant' in the market, with over 500,000 workers it offers anyone with a computer and internet connection the ability to 'earn $$$ while working from home!'. Requestors (companies or individuals) can then submit a HIT(Human Intelligence Task) to the mechanical turk system, this HIT is then displayed to all users along with a reward for its completion, the results are then sent back to the requestor and they decide whether the work is worth paying for. This system raises a problem though, How do I know how accurate my results are? You have no control over which worker selects your HIT and your task of 'Transcribe this podcast' could be carried out by a lacklustre employee in India who doesn't speak very good English. The solution most requestor guides offer is to submit your job to multiple workers but this pushes up your costs linearly and leaves you to make a difficult decision to make on the number of workers to consult.

We seek to create is a solution to this problem by providing a framework in which requesters can submit jobs with a required level of accurary. Our system then decides how many users it needs to consult to achieve this based on their expertise.

Contents

1 Executive Summary

2 Introduction

2.1 An Informal Discussion of Crowdsourcing and the problems it faces

Our project concerns *crowdsourcing* 'the book' would describe crowdsourcing as the principle of obtaining an accurate and appropriate result by having many different contributors performing a task but we'd like to start with a story which we believe encapsulates the idea of crowdsourcing in its most positive and useful light.

In January 2009 Timothy Gowers (Fields Medal winner and avid blogger) used his blog to post a striking question *Is massively collaborative mathematics possible?*. He posted a difficult and unsolved mathematical problem he was particularly interested in and invited people to contribute to its solution in the comments section. The project initally got off to a slow start but once the ice was broken the comments flooded in. 37 days , 27 contibuters and 800 comments later Timothy Gowers was able to announce that not only had they solved the original problem but they had also solved a harder generalisation of it, he called his experiment the *Polymath Project* .

This is a nice example of how crowd sourcing can be used to combine the skills of many individuals and produce answers to a complex problems, but there are many motivations to outsource your task to the crowd:

- *Computational Difficulty:* Timothy Gowers provided a nice example of a computationally difficult problem It is extememly unlikely you would be able to write a Java program or use Wolphram Alpha to produce a complex mathematical proof, some problems require what we like to call the 'human touch'. Problems which require the human touch are in no way confined to the realms of complex mathematics identifing an unknown bird in a picture for example would prove quite difficult on a computer you may need access some a program like Google Goggles but even then you would probablly need a good quality photograph, whereas one avid bird enthusiast in the crowd might be able to easilly identify the bird and reuturn the correct answer.

- *Saving Time:* In 2009 aviator Steve Fosset crash landed deep in the Nevada desert, his friends knew they had a very small and time critical window to find him alive and they had little faith in the current search and resuce operations. They organised for satellite images to be taken of the desert and the images were passed to a crowd who were asked to identify foreign object which could be potential crash sites. This is a nice example of how the crowd can be used to literally cover a large ammount of ground in a small ammount of time. Not all examples of saving time are quite this dramatic though, image tagging is an extemely arduous task for an individual or small group of people to perform and tagging a relatively large set of images could take weeks or even months, outsourcing this to the crowd could have the job done in a number of hours.

- *Saving Money:* Time is money as they say and this goes hand it hand with the point above. If you have to pay a team of high salary computing professionals to tag images for your project when you could be paying them to write code this is not cost effective, however passing this job off to a crowd of lower paid people could potentially save you a lot of money.

- *Reaching A Willing Audiance:* Unless it's something they enjoy the fact is that people are not willing to work for free, this is why getting the general public to do things such as complete surverys can be difficult as a large number of people will simply not want to do it. The crowd memebers will be inventivised to perform work and as such will be more likely to complete your survey.

This is all well and good in theory but it leaves us with a number of problems, an introduction to these problems is provided below but they are discussed and addressed in much greater details at various stages throughout the report:

1. *How do we get these problems to the crowd?*
   It is unlikely many people with a problem to be solved would want to go to the trouble of creating a crowd themselves as this would be comparitive to the complexity of the problem itself therefore there is a need for a third party crowd management system.

2. *What problems can we ask the crowd?*
   Specialised crowds have been successful and certainly have their uses for instance www.stackoverflow.com can be thought of as a crowd specialising in the solution of computing problems, however it is unlikely I would be able to find a specialised crowd to indentify bird pictures or to search satellite images for crash sites, therefore I need access to a generalised crowd able to adapt to and solve a wide variety of problems. The crowd management party therefore needs to provide the ability to ask a wide variety of questions and the ability to easily encorporate new questions in response to new technology.

3. *How many people do we ask, who do we ask and how can we trust what they say?*
   Crowd members will be a representitive sample of the general population, some will be brighter than others and some will be willing to put in more effort than others based on this you can place all annotators on a scale of trustworthiness which rates how much you believe an answer they give you. This rasises the question of 'how many people do I ask?', is consulting a small number of very trustworthy people better than a large number of non trustworthy people?, However I can't simply ask the same subset of people over and over again as workload will build up and my answers will be delayed. If I have a 'specialist' question do I direct it to someone with knowledge of that specialism? Clearly a sophisticated algorithm is needed on the crowd management side to address these problems.

A solution to these problems provides us with the basis for our project we are to:

Design and implement a framework that allows the optimum number of contributors to be selected on the basis of their trustworthiness for a desired accuracy of the outcome/result and that evaluates the trustworthiness of contributors on the basis of the accuracy of the results they provide.

2.2 A Formal Specification of Our Objectives

2.3 A Formal Discussion of Our Achievements

3 Design and Implementation

3.1 Data Processing

Once clients have submitted their tasks and the crowd are submitting answers, our goals are:

1. To estimate the correct annotation for each subtask based on crowd annotations and their respective likelihoods of being correct.

2. To use crowd annotations to better our understanding of the accuracy of a crowd

These goals can be achieved with the 'Expectation-Maximisation algorithm'. We have adopted this algorithm for the processing of annotations, generalised it to work over a variety of task types and implemented it to function efficiently in an online environment.

3.1.1 EM-Algorithm overview

The a EM algorithm models the system as a hypothesis, h, and new evidence, Y, which is a set independent instances with observed data, X, and unobserved data, Z.

It then iterates through two steps to revise this hypothesis so that it converges on the most likely hypothesis for the state of the system.

Expectation Step: You calculate P(Z) values using the X assuming the current hypothesis holds. In our case Z are the correct annotations for each subtask, X are the provided labels and the current hypothesis is the annotator‘s accuracy. [1]

$p(z) = p(z|\zeta) \prod p(l_{ij}|z_i a_j)$ (1)

where $p(z|\zeta)$ is our prior belief of $p(z)$.

Maximisation Step: You replace the current hypothesis h with the most likely hypothesis assuming that the Z = E[Z]. This will mean that we have to calculate our most likely accuracies for the system based on the expected correct annotations calculated from the previous step.

Q(a,a) (2) Annotation types

Our framework deals with a variety of subtasks that have different annotation types and models for accurate behaviour.

We model each of the subtasks with a different conjugate prior and model for accurate behaviour.

Subtask type, Description, conjugate prior, likelihood Binary: An annotator can respond either true or false Multivalued: An annotator must categorise the subject into one of a range of categories D Continuous: An annotator must respond using a tuple from an N-dimensional number line [2] f(1)

Online implementation

The online implementation of this algorithm is different in that it does not compute the expected labels as a batch job but instead has to observe several instances, one at a time, over a long period of time.

It also exploits discrimination. Expert annotators are asked first to respond to subtasks in order to reduce the total number of labels required. Since experts are likely to be the minority this technique is important since it makes sure that their annotations are taken into account [3].

We carry this out by checking the variance of our estimate for the most likely accuracy of an annotator. A large variance would suggest that we do not yet have enough information to judge an annotator. When the variance is small enough, we will test the accuracy against the criteria below.

Subtask type, criteria Binary: Multivalued: Continuous:

E-step Db response > Update ratios > check threshold & max labels >

M-step Update accuracies > check threshold variance > update bot/expert lists

E step

To calculate (1), values for the Sequential Probability Ratio Test [4] for each hidden value Z are stored in the database. Since the value is the sum of a sequence, the stored value can simply be increased by the new value. At this point each value is tested with the threshold to see if the maximum likelihood is great enough. At this point the M step will begin.

M step

To compute the new accuracy (2), the prior accuracy needs to be modelled as a distribution stated in figure 1 and the expected accuracy is calculated assuming the correct label is the most likely one from the last step.

The new accuracy would be the peak of the equation (2). To avoid using numerical methods to estimate the peak we write the Bayes estimate as a convex combination of the prior and the data [5]. This way we get a formula for the expected accuracy that can be computed in constant time:

Subtask type, convex combination Binary: Multivalued: Continuous:

Once the accuracies have been updated we must check the variance of the posterior accuracy. In the cases where the accuracy is modelled as a single beta distribution, multi valued and continuous, is trivial:

In the binary case the accuracy is modelled as a product of two dependant distributions therefore the variance must be estimated. We fit the peak to a multivariate Gaussian curve to estimate the posterior$\tilde{O}$s variance.

If we are confident in our knowledge of the annotator we test its accuracy against the criteria and store any new data about experts and bots in the database.

[1] Đ Machine Learning Tom M Mitchell [2] - A Compendium of Conjugate Priors - John D. Cook [3] Đ Online crowdsourcing: rating annotators and obtaining cost-effective labels - Welinder Perona [4] - Sequential tests of statistical hypotheses Đ Wald [5] Đ Bayesian inference for simple problems - Simon Jackman