Towards Task Recommendation in Micro-Task Markets

Vamshi Ambati, Stephan Vogel and Jaime Carbonell
Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213

Abstract

As researchers embrace micro-task markets for eliciting human input, the nature of the posted tasks moves from those requiring simple mechanical labor to requiring specific cognitive skills. On the other hand, increase is seen in the number of such tasks and the user population in micro-task market places requiring better search interfaces for productive user participation. In this paper we posit that understanding user skill sets and presenting them with suitable tasks not only maximizes the over quality of the output, but also attempts to maximize the benefit to the user in terms of more successfully completed tasks. We also implement a recommendation engine for suggesting tasks to users based on implicit modeling of skills and interests. We present results from a preliminary evaluation of our system using publicly available data gathered from a variety of human computation experiments recently conducted on Amazon's Mechanical Turk.

Introduction

Crowdsourcing has become popular in the recent years where one party can broadcast tasks on the internet to a large group of users that can compete and complete them for a micro-payment. Traditionally these tasks were performed by a resident employee or a contractor with a specific area of expertise. With crowdsourcing, such tasks are requested from an anonymous crowd, which is a mixture of experts and non-experts. The nature of the tasks is such that they are typically hard for computers to solve, but only require a few seconds for a human to complete. For example, identifying a person in a photograph, tagging a video for a particular event etc, flagging an email for spam, spotting characters in an image etc.

More recently, we see a trend of complex tasks being crowdsourced as well. Researchers from various fields of Science like Computer Vision, Natural Language Processing, Human Computer Interaction etc are embracing 'crowdsourcing' for acquisition of annotated data (Snow et al. 2008; Kittur, Chi, and Suh 2008). Table 1 shows a brief sample of the tasks posted on MTurk. The expectation of the requesters is that the large number of users in the crowd would offer a higher chance of finding a sufficiently skilled person to complete the task.

Field	Sample Tasks
Language	Translation, search relevance, grammar check, syntax annotation
Speech	dialog analysis, transcription
Vision	Semantic tagging of images, videos
HCI	Collaboration, Design surveys
Sociology	Decision theory, Inference
Biology	Annotating proteins
Miscellaneous	Reviewing, content creation

Table 1: Sample tasks in micro-task markets

Workflow in Micro-Task Markets

A number of micro-task markets have come into existence in the recent years. A micro-task market is a platform that enables the exchange and interaction of requesters of work and the labor. In this work we will refer to Amazon Mechanical Turk (MTurk) as an example platform for micro-task markets, but some of the observations in this paper are extendible to other similar platforms as well like CrowdFlower[1], Odesk[2] etc. Amazon Mechanical Turk (Mturk) is an online marketplace that enables computer programs to work with with humans via crowdsourcing. As shown in fig 1 requesters can pose tasks known as HITs (Human Intelligence Task), and workers, also known as turkers, can then browse among existing tasks or search using keyword to find and complete them. The requesters then receive the output, which they verify and approve payment upon satisfaction.

The Problem

We observe that this workflow is sub-optimal for both requesters and the workers. From the requesters perspective,

[1] http://www.crowdflower.com
[2] http://www.odesk.com

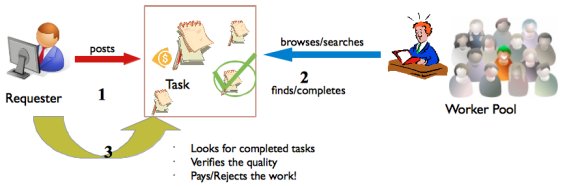

Figure 1: MTurk Workflow

they are constantly working on better approaches to fight with quality of data obtained on MTurk due to less skilled labor and spammers. The requesters also have to devise creative strategies to stay visible on the top pages of MTurk in order to be noticed by the crowd.

A recent study on search behavior in micro-task markets like MTurk (Lydia Chilton 2010) shows that workers look mostly at the first page or the first two pages of the most recently posted tasks. Therefore from the perspective of a worker, he is constantly working on some task that he may not be interested in or skilled at, but that appears attractive due to the reward. Very few users spend time to carefully browse deep into the first ten or more task pages to select the tasks that they wish to work on.

To summarize, we list the drawbacks caused by the current workflow in micro-task markets which does not seem to be working for either the requesters or the workers.

- Genuinely skilled workers who are more suitable for particular tasks may not be able to search and find them at the right time before the rest of the crowd and so can not contribute.
- Lesser-skilled workers may attempt the tasks and produce sub-standard or noisy output requiring the requesters to put in extra effort to clean and verify the data.
- Researchers may also spend more money than necessary by having to repost the tasks for eliciting redundant judgments.
- Less qualified workers may provide low quality and risk being rejected and in turn hurt their reputation in micro-task markets.
- A vicious cyclic effect leads to a market of lemons where requesters lack trust in workers and do not pay the right monetary rewards, attracting low quality turkers.
- Researchers and their underlying systems will be broken by bad input and ultimately discouraging them from using crowdsourcing.

Proposed Solution

Our hypothesis is that the current workflow in micro-task markets is a main contributing factor to the low-quality output that requesters observe. We therefore propose a recommendation engine that suggests work to a person based on skillset and interests. We posit that learning such interests is possible with the amount of data available in the form of implicit or explicit feedback provided by a user. We discuss the kind of information that is useful for building such a user preference model and also propose two different ways in which one can build recommendation engines.

Finally our position is that, as MTurk style markets grow in number of tasks and labor in the coming years, it will become difficult for users to find work using standard browse and search methods. A push mechanism in the form of a recommendation engine will be necessary and will lead to improved productivity of the workers. The requesters will receive better quality output as the tasks will be recommended or routed to the right workers. Eventually, the requesters can also reward the workers appropriately as they no longer have to deal with less skilled workers and spammers or spend on redundant annotations.

User Modeling

Obtaining information of the user is the key to user modeling. Information can be obtained in various methods. We list below such diverse kinds of information that are typically available to micro-task markets like MTurk. Most of this information is currently available to either the requester or the platform owner or both, and in some cases such information is easy to obtain although not currently available.

Profile

This is information about a user typically obtained in a structured manner as part of a sign-up process. Additional data about the user can also be obtained individually for each task. For instance, it is now a common practice on MTurk to get more information about the turker like location, country, time zone, education qualification etc. In some cases, such information can be implicitly obtained by tracking geolocation based on IP address.

Explicit Feedback

Ideally the data required for learning user preferences reliably is explicit ratings provided by the worker. More information can be extracted from the worker on different lines - how much they like a task, whether the reward is sufficient or whether the time available is sufficient etc. Requesters have started to design these extra questions into their tasks to better understand their workforce.

Implicit Feedback

While explicit feedback is more desirable for a reliable reflection of the user interests, it comes at the cost of user time and a risk of taking away the user focus from the original task. Implicit feedback on the other hands refers to information acquired through understanding the user actions in the micro-task market. For example, a user search query is indicative of his interest in a particular task, as is a click on the task link and completion. We therefore suggest extracting and using implicit feedback as much as possible.

Details of the Task From implicit feedback provided by user through interactions on the task, we can then accumulate additional information about the task. Each task posted on MTurk and other platforms is associated with meta information like below:

- Description of the task in the form of title, instruction set and keywords can cumulatively provide a better understanding of the expertise required for completion of the task. However, being expressed in natural language, it requires parsing and analyzing the text.
- Reward associated with the task also motivates and appeals to the user and so it is important to use as additional features while modeling interests of a user.
- Number of hits available for the current task also influences the choice of the user. There is evidence from existing literature that users on MTurk tend to select tasks that have a large number of associated HITs (Lydia Chilton 2010).
- Timestamp of the posted task can act as a deciding factor as to which of the participants it would interest.

Requester Feedback Perhaps the most informative of the implicit feedback is a successful completion of the task to the requester's satisfaction that begets payment. Rejection or success on a task as judged by the requester is a key information that reveals the expertise of the user. Bonuses, comments and other feedback from the requester, currently available as features on MTurk platform, are also cues that can be associated with the skill level of the user for the given task.

Learning A User Preference Model

In this section we propose two different methods for learning a user preference model based on information about the user collected as discussed in previous section.

Bag-of-Words Approach

Given the history of a specific user, $H = \{(t, c)\}$ and learn preference models from it, where t is the task and its associated features and $c \in 1..K$ to indicate the scale of preference of the user for the task on a scale of 1 to K. As we use implicit feedback in this work we only consider a binary preference, and therefore $c \in \{-1, 1\}$. The bag-of-words approach uses the vocabulary of the task description and computes the similarity as the overlap in their vocabularies.

$$bow(t) = \frac{1}{|H|} * \sum_{i=1}^{|H|} c_i * |(Voc(t) \cap Voc(t')|$$

Classification Based Approach

The bag-of-words approach can not incorporate the other features of a task like timestamp, reward etc. We therefore also propose using a binary classification based approach. We use a maximum entropy classifier for classifying between two classes $c \in \{-1, 1\}$, where $c = 1$ indicates a user will be interested in the task and $c = -1$ otherwise. The positive examples to train the classifier are all the tasks that a user has completed in the past, and for negative examples we select an equal number of tasks that the user has not attempted so far. The classifier probability can be defined as:

$$Pr(c_i|t) = \frac{1}{Z(t)} exp\left(\sum_{j=1}^{n} \lambda_j f_{ij}(c_i, t)\right)$$

where t is a task and associated features, c_i is the class, f_{ij} are feature functions and $Z(t)$ is a normalizing factor. The parameters λ_i are the weights for the feature functions and are estimated by optimizing on a training data set.

Number of different Tasks	114
Total HITs from all tasks	178,345
Unique Turkers	5,345
Turkers attempting 10 kinds of tasks or more	24

Table 2: Statistics of our dataset

Re-ranking for Recommendation

Given a list of tasks, the user preference model learnt above can be used to re-rank them and select the top few of the tasks for recommendation. For the bag-of-words based preference model, we can re-rank the list based on the similarity score. Similarly, for the classifier model, the posterior distribution probabilities can be used as a direct score for sorting the tasks.

Preliminary Experiments

Data Collection

The data required for learning such models require explicit user preferences, which can be collected by conducting a survey or implicitly by gathering information from user interaction on MTurk. When a user clicks on a task and attempts to complete the task we assume that the user is interested in the task. This is similar to the pseudo-relevance feedback concept used popularly in information retrieval. However, even this kind of data is difficult to obtain as it is present only with the requester of the data. The recent NAACL 2010 workshop on crowdsourcing has made publicly available all the data collected as part of the workshop [3]. The data was collected as part of a month long effort from multiple requesters seeking data for a diverse variety of tasks. The table 2 below provides some statistics about the dataset.

Evaluation

We can evaluate the performance of a recommendation engine by the number of times the suggestion was liked by the user, as given by direct or indirect feedback. From our dataset we found 24 users that have attempted more than 10 different kinds of tasks. For each such user we use half of the instances to train the user-specific models, and then re-rank the remaining set of tasks using the model. The evaluation metric we use is a 'precision@N' which is a metric in Information Retrieval that calculates the number of times

[3] http://sites.google.com/site/amtworkshop2010/data-1

Approach	@1	@2
Similarity + Description	58.33	79.16
Maxent + All features	54.16	70.83

Table 3: Evaluation of Re-ranking Tasks

a preferred result is seen in a subset of 'N' retrieved documents. Here we re-rank the tasks and compute '@1' and '@2' for all the 24 users and report the ratio of users where the recommended task was in deed completed by them.

Since we conduct our analysis on dataset collected from already completed MTurk experiments, we can verify whether the suggested task was completed by the turker. Results from our preliminary experiments can be seen in Table 3 and show that we can in fact suggest tasks that are potentially interesting to workers based on their previously completed tasks. The classification based approach to recommendation underperforms the similarity based approach due to the extremely sparse data scenario, but we hypothesize that it will perform better in a real-world scenario. Also, manual inspection on some tasks shows that when such users complete the suggested tasks, the quality is comparable to that obtained from repeated labeling.

Our dataset was collected within a period of time and we conduct re-ranking of this list of tasks. Therefore the evaluation using such a dataset may only reveal the precision of our recommendation engine and ignore recall completely. We do not have information about all the tasks that the user may have completed on MTurk or the other tasks that were available at that point in time. However, this is the only available dataset currently that is suitable for our work and so the results mentioned here are only indicative of the effectiveness of a recommendation system.

Related Work

The motivation for our work is the observation of drift in the nature of tasks in micro-task markets from simple to complex. Recent work also supports this observation. (Haoqi Zhang and Parkes. 2011) suggest the need for expert inference and routing the tasks in order to solve complex problems. Similarly (Aniket Kittur 2011) is a framework for breaking complex tasks into smaller tasks that can be completed through crowdsourcing. In our current work we actually implement and evaluate methods for user modeling and task recommendation and therefore such frameworks can benefit immensely from our work of automatically identifying the experts.

Redundant labeling using multiple annotators has been a well known strategy to deal with non-expert and noisy annotators. (Snow et al. 2008) propose worker modeling as a form of bias correction in crowd data. They estimate worker models computing accuracies on a gold standard dataset. (Sheng, Provost, and Ipeirotis 2008) propose repeatedly obtaining multiple labels for a sub set of the data to improve overall label quality. (Donmez, Carbonell, and Schneider 2009) propose a novel way of dealing with multiple noisy translators with varying cost structures and accuracies. While prior work approached quality by identifying right label from multiple labels, our work aims to improve quality by routing the tasks to the right workers.

Ongoing and Future Work

We list below some interesting research challenges that we can arise from the work proposed in this paper and some of which we are currently working on.

- A larger dataset of tasks is needed for observing the effectiveness of recommendation. Such real world data can only be obtained from MTurk or the other micro-market operators. This will significantly help the work, similar to how the Netflix datasets have helped understand recommendation engines.
- New users do not have profile information on MTurk and it would be interesting to bring in techniques from collaborative filtering to complement sparse data scenarios.
- Generalizing on the feedback provided by users will be a key component in reducing the amount of feedback a user has to provide on an ongoing basis.
- We need new features that look beyond just the meta-level task descriptions, but also factor the actual content of the task. For example, "'review an essay" is a title of a task that could potentially be interesting to a user, but understanding the content and topic of the essay can help us select a more appropriate worker.
- Incorporating user specified constraints like cost, time into the re-ranking. E.g a user with only 15 minutes of spare time needs to be shown interesting tasks for the stipulated time constraint.

References

Aniket Kittur, Boris Smus, R. E. K. 2011. Crowdforge: Crowdsourcing complex work. Technical report, Human-Computer Interaction Institute, School of Computer Science, Carnegie Mellon University.

Donmez, P.; Carbonell, J. G.; and Schneider, J. 2009. Efficiently learning the accuracy of labeling sources for selective sampling. In *KDD*, 259–268.

Haoqi Zhang, Eric Horvitz, R. C. M., and Parkes., D. C. 2011. Crowdsourcing general computation. In *CHI 2011 workshop on crowdsourcing and human computation*.

Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with mechanical turk. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 453–456. New York, NY, USA: ACM.

Lydia Chilton, John Horton, R. M. S. A. 2010. Task search in a human computation market. In *Human Computation Workshop*.

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 614–622. New York, NY, USA: ACM.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the EMNLP 2008*, 254–263.