Expert Systems with Applications 41 (2014) 6190–6210

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Dynamic estimation of worker reliability in crowdsourcing for regression tasks: Making it work

Alexey Tarasov *, Sarah Jane Delany, Brian Mac Namee

School of Computing, Dublin Institute of Technology, Kevin St, Dublin 8, Ireland

A R T I C L E I N F O

Keywords:
Supervised machine learning
Crowdsourcing
Worker reliability
Multi-armed bandits
Emotion recognition

A B S T R A C T

One of the biggest challenges in crowdsourcing is detecting noisy and incompetent workers. A possible way of handling this problem is to dynamically estimate the reliability of workers as they do work and accept only those workers who are deemed to be reliable to date. Although many approaches to dynamic estimation of rater reliability exist, they are often only appropriate for very specific categories of tasks, for example, only for binary classification. They also can make unrealistic assumptions such as requiring access to a large number of gold standard answers or relying on the constant availability of any rater. In this paper, we propose a novel approach to the dynamic estimation of rater reliability in regression (DER^3) using multi-armed bandits. This approach is specifically suited for real-life crowdsourcing scenarios, where the task at hand is labelling or rating corpora to be used in supervised machine learning, and the annotations are continuous ratings, although it can be easily generalised to multi-class or binary classification tasks. We demonstrate that DER^3 provides high-accuracy results and at the same time keeps the cost of the rating process low. Although our main motivating example is the recognition of emotion in speech, our approach shows similar results in other application areas.

1. Introduction

Supervised machine learning often makes use of human workers (raters) to collect target ratings for training data. For example, in a spam filtering scenario raters would mark a collection of e-mail messages as either spam or non-spam, and this collection would be used to train a spam recognition model. In order to get accurate ratings, several raters are often asked to rate each instance from the data. These ratings are then aggregated in some way—for instance, by taking a mean or a majority vote—to produce a single rating for each instance. Although multiple raters have been widely used in many areas, including labelling volcanoes on the images of the surface of Venus (Smyth, Fayyad, Burl, Perona, & Baldi, 1995), machine translation (Ambati, Vogel, & Carbonell, 2010) and sentiment analysis (Brew, Greene, & Cunningham, 2010), such a setup has one challenge: the inevitable presence of unreliable raters. Noisy raters can prolong the rating process and lead to inaccurate ratings (Sheng, Provost, & Ipeirotis, 2008). One of the possible solutions to the problem of noisy ratings is to use a large number of raters to rate every instance as suggested by Sheng et al. (2008), but such an approach can also significantly increase the overall cost of the rating process. Finding a balance between error and cost is one of the challenges in modern crowdsourcing scenarios. According to Welinder and Perona (2010), a sensible approach to this problem is to autonomously discover, and disqualify, unreliable raters as early as possible in the rating collection process. It can significantly improve the quality of the ratings gathered and keep costs to a minimum. We refer to this as the *dynamic estimation of rater reliability* and this is the problem addressed by this paper.

Dynamic estimation of rater reliability has been studied extensively over the last few years (Donmez, Carbonell, & Schneider, 2009; Welinder & Perona, 2010; Yan, Rosales, Fung, & Dy, 2011). However, such techniques usually make unrealistic assumptions, for instance, requiring access to the gold standard against which the rater answers can be compared (Ho, Jabbari, & Vaughan, 2013). Also, the state-of-the-art in dynamic estimation of rater reliability tends to concentrate on binary classification (Dekel, Gentile, & Sridharan, 2012; Ho et al., 2013), although multiple raters are also often used in multi-class classification or regression tasks. One more common limitation is that a lot of dynamic techniques require knowledge about the task such as the statistical distribution of rater errors (Welinder & Perona, 2010) before the rating process can begin. Other approaches need a set of features to be supplied with every training instance (Dekel et al., 2012; Yan et al., 2011). Finally, many of the existing approaches assume that every rater is available to rate at the time when his rating is identified as being needed (Tran-Thanh, Stein, Rogers, & Jennings,

* Corresponding author.
E-mail address: tarasovs.aleksejs@gmail.com (A. Tarasov).

http://dx.doi.org/10.1016/j.eswa.2014.04.012
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

2012; Wu, Liu, Guo, Wang, & Liu, 2013). This might be the case in certain conditions, but when the rating process is conducted using a crowdsourcing service such as the widely used Amazon Mechanical Turk[1] or Crowdflower[2] services, raters can start and finish rating at any time and often may be unavailable to rate when their rating is deemed needed.

Although existing dynamic techniques make different assumptions and work under different conditions, all of them solve the problem of finding a balance between *exploration* (giving all raters a chance to rate in order to estimate their reliability precisely) and *exploitation* (using the best raters discovered). This task can be efficiently solved as a *multi-armed bandit* (MAB) problem (Robbins, 1952) which represents the task at hand as a multi-armed gambling machine.

In this paper we propose DER^3, a practical MAB-based approach to the dynamic estimation of rater reliability in regression which does not have most limitations of the state-of-the-art approaches. Namely, DER^3 (i) is particularly suited for regression, although, can be applied to a variety of tasks including binary and multi-class classification, (ii) does not require any previous knowledge about the task, (iii) assumes that the pool of raters is not fixed in advance, and any of them can perform as much work as they want at any time, (iv) does not require a set of features associated with instances and (v) works in the conditions when the quality of a single rating cannot be evaluated independently. Our primary application area is automatic emotion recognition from speech; however, additional evaluation on datasets from other application areas shows that our approach also works well in other domains.

The paper is structured as follows. Section 2 covers related work in the estimation of rater reliability as well as providing an overview of MABs and how they are used in multiple-rater scenarios. Then we propose and evaluate MAB-based approaches to dynamic estimation of rater reliability. We do it in two steps: first, we evaluate whether MABs can be used to estimate reliability precisely. In order to do this, we use MABs in a simplified scenario, namely, assuming that the pool of raters is fixed in advance and every rater is available to rate immediately (Section 3). In Section 4 we relax this assumption, formulate the DER^3 approach and show that MABs can be used in real-life crowdsourcing conditions, when raters can become available at any time. Section 5 concludes the paper and suggests directions for future work.

2. Related work

Usually rating by multiple raters happens via *crowdsourcing*, “a type of participative online activity in which an individual, an institution, a non-profit organization, or a company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task” (Estellés-Arolas & Gonzalez-Ladron-de Guevara, 2012). The notion of a “flexible open call” means that this definition includes not only scenarios where the crowd of anonymous raters was hired through a platform such as Amazon Mechanical Turk, but also cases when the call was limited to a community with a specific knowledge or expertise (Whitla, 2009) such as medical experts, speakers of a certain language or customers of a specific company.

When used in supervised machine learning, the crowdsourcing process usually requires every instance in a specific dataset to be rated by several raters. There usually is a certain budget, from which a fixed payment is paid for each rating collected. Involving multiple raters in rating each instance can make crowdsourced solutions quite expensive (Ipeirotis, Provost, & Wang, 2010). That is why the task of decreasing the overall cost of the rating collection receives a lot of attention (Donmez et al., 2009; Welinder & Perona, 2010; Yan et al., 2011).

When all ratings are gathered, they are aggregated to provide *a prediction*, a single answer for every instance. Then these predictions are used as a training set (to train a classifier/predictor) or as a validation set (to measure the performance of a classifier/predictor that has already been trained). It is expected that the predictions are close to the *gold standard*, a set of true ratings for each instance that are not known in advance. A large volume of research reports that these predictions are indeed quite accurate (Nowak & Ruger, 2010; Paolacci, Chandler, & Ipeirotis, 2010). Applications where crowdsourcing was successfully used for rating tasks include computer vision (Welinder & Perona, 2010), natural language processing (Snow, O'Connor, Jurafsky, & Ng, 2008) and medical imaging (Raykar et al., 2010).

A typical scenario of collecting ratings is the following: every rater can rate a single instance once, and all raters do exactly the same task: provide a rating when being presented with an instance without interacting with other raters (Donmez et al., 2009; Karger, Oh, & Shah, 2013; Raykar et al., 2010; Whitehill, Ruvolo, Wu, Bergsma, & Movellan, 2009). A few researchers present complicated multi-stage rating processes: for instance, Dai, Lin, Mausam, and Weld (2013) proposed a framework where answers can be iteratively improved. They used recognition of handwriting as one of the motivating examples. In such a setup each instance (a hand-written sentence or paragraph) is presented to a rater who can leave some of the words unrecognised. Such partial recognition can be a great help to a second rater, who might be able to recognise the previously unrecognised words by context. One other interesting exception is the work by Fang, Zhu, Li, Ding, and Wu (2012), who explored a model in which raters can teach each other.

Independent of the rating process details, there will always be noisy raters, who provide inaccurate ratings either because of a lack of expertise or in order to get payment without investing any effort. There are different quality control techniques that allow the detection of such inaccurate ratings and can eliminate them or compensate for them. Some of them are based on the previous performance of raters on other tasks; however, this information might not always be available, especially if crowdsourcing is used outside of a resource similar to Amazon Mechanical Turk. It is also possible to discover noisy ratings by looking at the time a rater worked on the task, but this technique is well known, and noisy raters can easily accommodate for it. More universal and reliable techniques are based on the actual results raters provide. These techniques can be divided into three groups, depending on the stage of the rating process at which they occur:

1. ***Before the start***: before a rater can rate any instances, he has to go through a qualification test (for example, rating a few test instances for which the gold standard is already known (Heer & Bostock, 2010; Su, Pavlov, Chow, & Baker, 2007)). If a rater fails this task, he is not permitted to rate any instances.
2. ***After the finish*** **(*static* estimation of rater reliability)**: any rater can rate as many instances as he likes. When all ratings are collected, a procedure is used to estimate rater reliabilities. When calculating predictions, ratings coming from the raters with high reliability have more weight than those coming from unreliable raters (Raykar et al., 2010; Whitehill et al., 2009).
3. ***During the process*** **(*dynamic* estimation of rater reliability)**: the reliability of raters is tracked dynamically as they rate instances. As soon as an unreliable rater is detected, he is not presented with new instances to rate (Donmez et al., 2009; Welinder & Perona, 2010).

[1] www.mturk.com.

[2] www.crowdflower.com.