Feature Factory:
Crowd-sourced feature discovery

Kalyan Veeramachaneni
CSAIL, MIT.
32 Vassar Street
Cambirdge, MA 02139 USA
kalyan@csail.mit.edu

Una-May O'Reilly
CSAIL, MIT.
32 Vassar Street
Cambirdge, MA 02139 USA
unamay@csail.mit.edu

Kiarash Adl
CSAIL, MIT.
32 Vassar Street
Cambirdge, MA 02139 USA
kiarash@csail.mit.edu

L@S 2015, Mar 14-18, 2015, Vancouver, BC, Canada
ACM 978-1-4503-3411-2/15/03.
http://dx.doi.org/10.1145/2724660.2728696

Abstract

We examine the process of engineering features for developing models that improve our understanding of learners' online behavior in MOOCs. Because feature engineering relies so heavily on human insight, we engage the crowd for feature proposals and guidance on how to operationalize them. When we examined our crowd-sourced features in the context of predicting stopout, not only were they impressively nuanced, but they also integrated more than one interaction mode between the learner and platform and described how the learner was *relatively* performing.

Introduction

We have been trying to quantitatively characterize learners' online behavior from *web logs* and *click stream* data. The raw data, after processing, curating, and storing in a database [1], enables extraction of *per-learner* time sequences of click stream events. These sequences are primitive but, if formulated into *variables* that abstract learners' behavior, via *feature engineering*, they could help gauge learners' intent, interest and motivation in the absence of verbalized or visual feedback. We are interested in:

[1] These three steps are extremely complex and challenging but are not in the scope of this paper

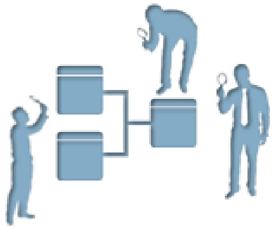

Figure 1: Engaging crowd to understand the data from Massive Open Online Courses.

Variables that capture per learner behavior with respect to a *resource*: For example, consider two variables such as: *total time spent of the video* and the *number of pauses while watching the video*. When these two variables are evaluated for all the learners and analyzed they can reveal patterns; if too many learners *pause* too many times, the video could be fast and/or confusing.

Per-learner longitudinal variables: For example, consider *time each student spent on the course website during the week* or, more complex, *on an average, the length of time before the deadline that the learner starts to work on an assignment*.

Humans need to help engineer features

Feature engineering learner variables should be primarily driven by humans, rather than be automated, because humans can:

Generate ideas based on their intuition: Everyone has been a student so it is possible to self reflect to invent variables. E.g, when considering prediction of *stopout*, we might each quite naturally suggest "*If the student starts to homework problems very close to the deadline, he might be very likely to fall behind and eventually drop out*". Subsequently, we might propose how to operationalize such a variable into a quantitative value by measuring, " Time difference between the deadline and the student's first attempt for the homework problem". While many other aspects of feature engineering can be automated, intuitive generation one cannot.

Offer instructor knowledge: For MOOCs, designing variables requires understanding of *context* and *content* of the course for which the variables are sought. This makes instructors or experts in the course perfectly positioned to propose variables. E.g., an instructor might be aware of an important concept whose understanding is critical for continued success in the course and may hypothesize that a variable that captures whether the learner understood the concept or not could help predict stopout.

Offer highly specialized learning science knowledge: Researchers from learning sciences are able to propose variables grounded in theory that elucidate latent constructs such as *motivation*, *intention*, and *self-efficacy*.

Help operationalize feature ideas: Moving from a feature idea to its operationalization is involved. Key decisions have to be made about definition and thresholds. For example, we might have to define what constitutes as "start" time for student working on an assignment. Since there is no mechanism where students notify when they started to work on the assignment, a human can help judge whether it should be the first time they looked at the problem, or the time of the first attempt for the problem or the time they attempted but saved the answer instead of checking for correctness.

Given that humans are NOT replaceable in feature engineering, we are exploring *how to increase the number of people who can participate in it*. Our exploratory context is predictors for *stopout*.

Naturally, we started by thinking up feature ideas ourselves then operationalizing them. Realizing this tact is vulnerable to missing some features, we have constructed activities that allow us to solicit feature ideas from others, i.e. the "crowd". This expands our feature set and eliminates our blind spots. Post-hoc, we can compare our original set to the crowd's set and discern whether it provides extra value.

Stopout prediction problem

Herein we define our notion of *stopout*. We considered defining it by the learner's last interaction in the course, regardless of the nature of the interaction [1]. However, this definition yields noisy results because it gives equal weight to a passive interaction (viewing a lecture,

	Describe feature	Why is this feature useful?
	pset grade over time: Difference between grade on the current pset and average grade over previous psets. Significant decreases may be likely to be correlated with dropouts.	Anecdotally it appears that users who perform poorly on the current week (especially after not performing poorly in the preceding weeks) will subsequently give up. They may also, with low probability, post on the forum explaining their issue with the task at hand.
	average pre deadline submission time: average time between problem submission time and problem due date.	people who get things done early are probably not under time pressure that would make them drop out.
	proportion of time spent during weekends): Fraction of observed resource time spent on each day of the week (7 variables for Mon-Sun that add up to 1). Just for previous week, and averaged over all weeks so far.	Heavy weekend users might be more likely to drop out, because they don't have spare weekday time to devote to the course.

Table 1: Three examples of features proposed by the students and instructors in the MIT class.

accessing an assignment, viewing a Wiki etc) as it does to a pro-active interaction (submitting a problem, midterm, assignment etc). A learner could stop submitting assignments in the course after week 2, but continue to access the course pages and not be considered stopped out. Instead, we define the *stopout*point as the time slice (week) a learner fails to submit any further assignments or exercise problems. A submission (or attempt) is a submission of any problem type (Homework, lab, exam etc.). Using this definition for *stopout* we extracted the week number when each learner in the cohort stopped out. To illustrate the predictive model's potential application, we will use a realistic scenario. The model user, likely an instructor or platform provider, could use the data from week 1 to week 3 (current week) to make predictions. The model will predict existing learner *stopout* during weeks $i+1$ to 14.

6.MITx Experiment

To generate ideas for features, we sought help from a class at MIT called 6.MITx. We presented the data model (what was being collected), explained what we meant by a feature and asked members of the class (professors and students) to posit features for each student/learner that could predict a student's stopout. We collected the input *via* a google form asking the users to describe each feature and describe why they think it will be useful in predicting stopout. We did not present our ideas for features to the class.

Outcomes: Out of the 30 features that the class proposed, 7 were in common with ours. Out of the remaining 23 features, we extracted 10. The features proposed by the students and instructors in this class were *intuitive*, based on *experience* and self identification as once/or currently being a student. Participants also gave detailed reason as to why the feature is useful. We present three examples in Table 1.

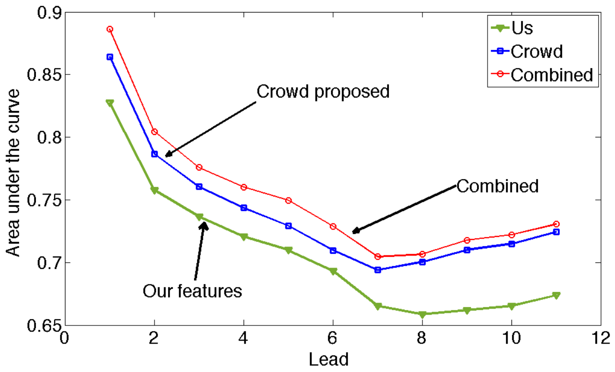

Figure 2: Performance of different sources of features in terms of prediction accuracy for stopout prediction problem at the end of week 3. We compare crowd proposed features with features proposed by and then performance when both of them are combined.

When we extracted these features and used them for stopout prediction, they consistently performed better than the features we ourselves came up with. In Figure 2 we compare the predictive accuracy (measured in AUC -higher the better) for different week-ahead prediction problems at week 3. We see that the features proposed by crowd help significantly, specially when we are trying to predict far ahead. Also note that combining the features proposed by the *crowd* and us leads to an even better performance. In general, for different learner cohorts we found that features proposed by the crowd mattered significantly more than the features we proposed ourselves.

We also found the more influential features were quite nuanced and complex. They incorporated data from multiple modes of learner activity (submissions, browsing and collaborations), required carefully linking data fields. Relational features that compared a learner to others and statistical summaries were proposed by the crowd and mattered quite a bit.

Based on this experience, we are building a web-based platform (WWW.FEATUREFACTORY.ORG) that allows many people to participate in defining features. The general public can enter a new idea, or comment on an existing one while programmers can script features.

Related work

The “more features the merrier" theme is prominent among feature engineering studies. For example, the 2010 KDD cup resulted in a paper "Feature engineering and classifier ensemble for KDD cup 2010" [2]. In the 2013 “Big data for education" MOOC offered by Prof. Ryan Baker, he suggests that, in practice, it is a process of ideation that happens by researchers brainstorming as a group with support for the free flow of ideas.

References

[1] Balakrishnan, G., and Coetzee, D. Predicting student retention in massive open online courses using hidden markov models. In *Technical Report No. UCB/EECS-2013-109*, EECS, University of California, Berkeley (2013).

[2] Yu, H.-F., Lo, H.-Y., Hsieh, H.-P., Lou, J.-K., McKenzie, T. G., Chou, J.-W., Chung, P.-H., Ho, C.-H., Chang, C.-F., Wei, Y.-H., et al. Feature engineering and classifier ensemble for kdd cup 2010.