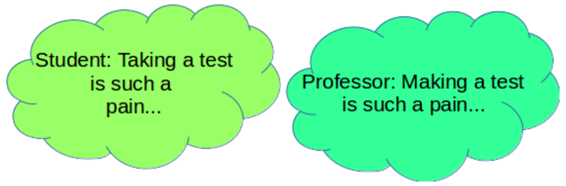
Automatic Assessment Generation via Machine Learning

Arjun Jauhari (Advised by Igor Labutov and Christoph Studer)

The Problem

No automated tools available that generate an assessment through which user can test his/her skills



- Available methods to generate/prepare a test requires an excessive amount of human (generally a teacher or domain expert) effort
- The traditional approach to assessment generation does not scale to test in multiple fields or even personalized tests

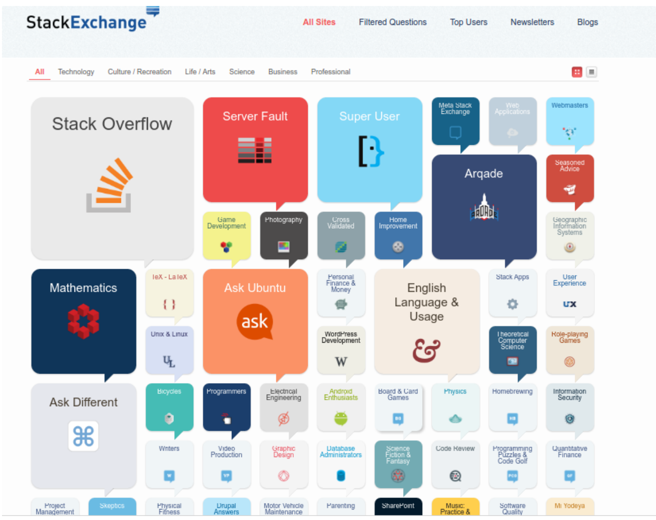
Proposed Solution

We use machine learning to automatically generate assessment using already available million of questions and answers from websites like stackoverflow.com

- No (or only little) human effort required and we can generate new assessments almost instantly and personalize them
- We can easily mix both theoretical and practical questions

Challenge: How to learn interesting hidden variables like difficulty of question, quality of answers, ability of users

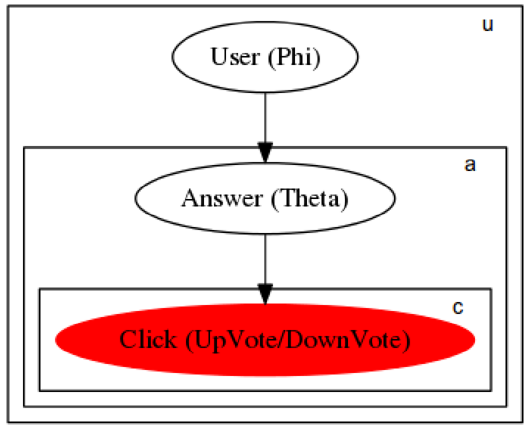
Dataset



- 130+ stackexchange websites, most famous one is stackoverflow.com
- Number of users = 5,277,830 ($\sim$ 5 million), file size 1.5 GB
- Number of posts (Question + Answers) = 29,499,662 ($\sim$ 30 million), file size 45 GB
- Number of votes = 98,928,934 ($\sim$ 99 million), file size 9 GB

Algorithm: Statistical model for answer quality and user ability

We used the following probabilistic graphical model to characterize the hidden parameters of interest

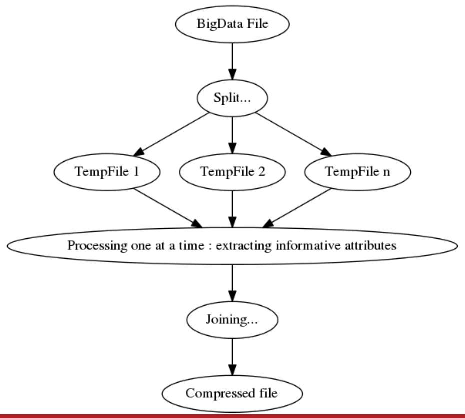


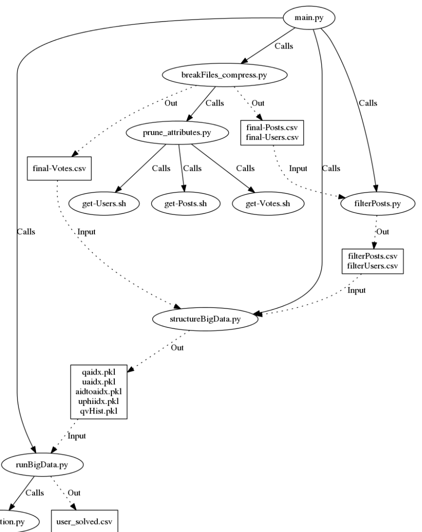
Parameters we learn are θ (Quality of Answer) & ϕ (Ability of User) Below equations model the conditional dependence

- $P(\theta,\phi|clicks) \propto P(clicks|\theta,\phi) * P(\theta,\phi)$
- $P(\theta,\phi|clicks) \propto P(clicks|\theta) * P(\theta|\phi) * P(\phi)$
- $Objective_{MLE} = \text{argmax}_{\theta,\phi} NLL(P(clicks|\theta) * P(\theta|\phi) * P(\phi))$, solved using Gradient Descent (AdaGrad and LBFGS)
- $P(click = k|\theta_1, \ldots, \theta_n) = \frac{exp(\theta_k)}{exp(\theta_1)+\ldots+exp(\theta_n)}$
- $P(\theta_i|\phi_j) \sim \mathcal{N}(\phi_j, \sigma^2)$
- $P(\phi_j) \sim \mathcal{N}(0, \sigma^2)$

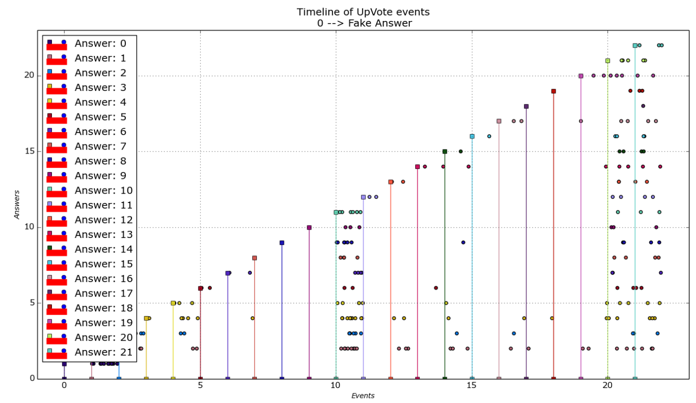
Pre-processing of Big Data & Implementation Details

To process such large files, we broke them into several smaller files and processed them separately



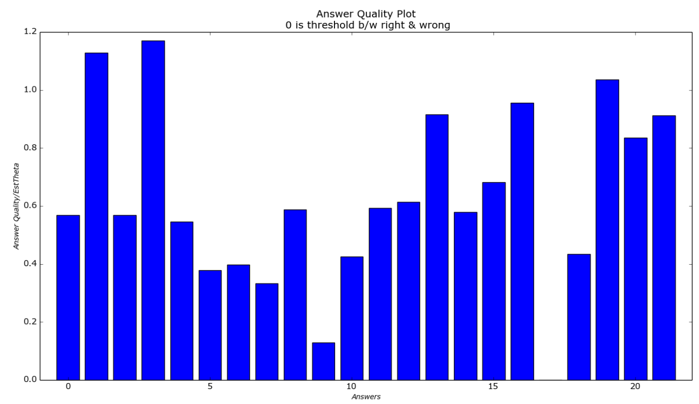


Time-line of a typical question in Real Data

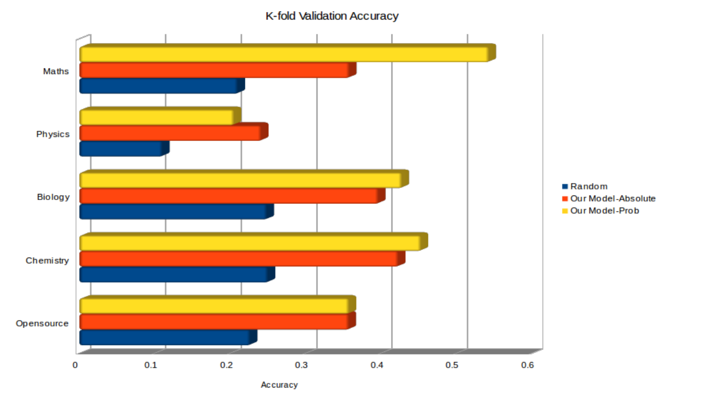


- Shows the entire vote and answer generation history of a particular question capturing the number of answers
- Each dot captures a click event (up-vote or down-vote)

We can learn the quality for every answer in above time line



K-fold Cross Validation Accuracy on Real Datasets



- We use K=10. We predict the clicks (UpVote/DownVote) made by Users and average over each fold.
- Note: Its a multiclass problem with varying number of classes.
- Evaluation Metric 1 : Absolute - $click = argmax(\phi_j; j \in X)$, X is list of answers for this question.
- Evaluation Metric 2 : Probabilistic - $click \sim P(\frac{exp(\phi_j)}{exp(\phi_1)+\ldots+exp(\phi_n)})$