

CS Educators Stack Exchange: Exploring User Interactions

Sukanya Moudgalya & K. Bret Staudt Willet

12/11/2018

Introduction

The Stack Exchange Network is one of the largest question-and-answer platforms in the world, founded in 2009 and boasting over 100 million unique monthly visitors by 2015. It has many subject-specific forums for coding related topics, Mathematics, Engineering, Photography, etc. The SE network has 133 different sites as of October 2018. More recently, Stack Exchange has recently opened up discussion forums for Math educators (2013) and Computer Science educators (2017) as well. For this project, we will focus on the users of the discussion forum Computer Science Educators Stack Exchange (CSEd SE). The description of the forum calls it “A question and answer site for those involved in the field of teaching Computer Science.”

In this forum, registered users can ask questions on a wide range of topics related to Computer Science (CS) education. Once they ask a question, they can assign it ‘tags’, so that similar groups of questions can be identified by a common theme or idea. Some tags that are currently present on the forum include ‘curriculum design, student motivation, algorithm’ etc. Users can also reply to (i.e., answer) questions posed by other users, comment on both questions and answers, upvote and downvote questions and answers, and so on through the voting mechanism, users are held accountable to what they post on the forum. The content on the forum is thus regulated by the forum users and members to a large extent.

In addition to ‘normal’ user regulation, a set of registered users also act as ‘moderators’ or ‘editors’. Editors can edit the text of what other people post. Moderators, on the other hand, have more power. They can follow up with posts that are flagged as spam or offensive, delete and manage other users, combine or collapse tags, lock posts so that no one can change them or vote on them, and so on (read more here).

The users on this forum range from actual K-12 and college-level CS educators to working professionals in the software industry, students, and other people interested in CS education. The users also belong to many different countries, such as the USA, UK, Canada, Ireland, India, Finland, etc. Each user has reputation points primarily based on the quality of their posts, such as user-generated votes their posts receive. The reputation changes across time are accessible for people to see.

Purpose

This study explores the kinds of interaction between the users, with a focus on the top voted questions. It seeks to describe the kind of users, based on their profession, location, reputation, etc., that tend to ask questions or respond to the top questions. It also seeks to understand if the users tend to cluster around certain question tags (topic area) or if the clusters are agnostic to the topics and user characteristics.

In the second part of the study, we will focus on trying to interpret the reasons for users choosing to answer the questions. It will explore if the reasons are dependent to the topic tag or some characteristic of the users who ask the question.

Theoretical Framework and Research Questions

In the US, many K-12 CS teachers are isolated as the only CS educators in their schools. They are often referred to as ‘singletons’. This isolation greatly impacts the opportunities of subject-specific peer-learning that these CS teachers might need (Yadav, Gretter, Hambrusch, & Sands, 2016). Peer-learning is a crucial component that teachers need for their learning and growth. Indeed, teacher communities and ‘professional learning networks’ have been extremely successful in the past for teachers’ social, cognitive, affective, and identity growth (Trust, Krutka, & Carpenter, 2016). In particular, there have been cases where high school CS teacher networks were central for inculcating a sense of community, promoting teacher reflection, and helping create a change in teaching practices (Ni, Guzdial, Tew, Morrison, & Galanos, 2011). As many CS teachers might have time constraints, busy schedules, and geographical limitations to form offline CS communities, it becomes important to also foster online communities for CS educators. It is no surprise that online networks for CS teachers have been growing in the recent years (Brown and Kolling, 2013). In this study, the online network we will focus on will be CSEd SE.

Given the context of the study, we choose to focus on the ‘community of practice’ framework to set a base. Communities of practice are “...groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly” (Wenger, 2011, p. 1). These communities have the following traits: (a) a domain of shared interest; (b) people who engage in joint activities and discussions; and (c) members who are practitioners (such as educators). Further, according to Wenger, they should demonstrate (a) mutual engagement (such as a dialogue instead of one directional information flow); (b) a joint enterprise; and (c) a shared repertoire. These three traits represent “...three dimensions of the relation by which practice is the source of coherence of a community” (Wenger, 1998, p. 72). The mutual engagement component is especially important because a one-directional sharing of information will not be representative of a community.

The CSEd SE has many elements that could potentially make it a ‘community of practice’. For instance, there are clear boundaries in terms of membership to the forum. The forum has a unique purpose, that is to build knowledge and dialogue in the domain of CS education. Further, the discussions on the forum are not generally a unidirectional venture. The commenting feature, for instance, allows scope for a back and forth dialogue. Finally, the presence of moderators and editors for maintenance of the forum, scope of democratic elections for the selection of moderators, reputation points based on user-generated voting system, gives CSEd SE the traits of a community of practice. Indeed, a literature review of informal online teacher communities revealed that the communities of practice framework has often been used to study teacher interactions in online forums (Macia & Garcia, 2016).

Given this framework, we will be exploring the directionality of answering questions, that is, explore the extent of mutual engagement, in CSEd SE. We will also be exploring if there are any clusters of users in the forum. There might also be some individuals who have a high between centrality, that users who might act as bridges between different sub-groups or clusters within the CSEd SE network. Identifying these ‘knowledge brokers’, who have the potential to help moving knowledge between different places and clusters, can be important for practitioners interested in PD for CS teachers (Wenger, 2000).

There might also be some users who have a high out-degree score, that is users who might be highly connected, have high popularity or reputation, and thus act as a source of high social capital and knowledge. Previous studies using Social Network Analysis on the Stack Exchange Network, in particular the Stackoverflow website, found that users with high reputation points tend to ask more questions and also answer more highly voted questions (Movshovitz-Attias, Movshovitz-Attias, Steenkiste, & Faloutsos, 2013). Further, activities of users in their initial stages of joining the forum have been proven to be predictive of their future activities. Thus, it might be possible to identify experts on these forums quite early. This identification of expertise might be beneficial to practitioners in fostering and designing discussion forums for educators.

The network studies on Stack Exchange Network until now have not focused on education related forums. In addition, they have used only reputation points of users to indicate the amount of good quality answers given by the users. In this study, we will also focus on whether the topic of the question and any prior relationship with the user who asked the question, influences the way a question is answered. By doing this, we hope to

see if the expert users are agnostic to the users or topics that they answer. On the other hand, users might also exhibit expertise in particular topics, by preferring to answer questions only in certain topics. This again might be beneficial for practitioners interested in developing online communities of practice for CS educators to identify topic-specific experts.

Using this communities of practice framework, three research questions guided our study:

- * *RQ1*. What are the characteristics of users who contributed to the CSEd SE’s core network of interactions?
- * *RQ2*. Does network clustering occur in the context of the CSEd SE’s core network of interactions? If so, what are some characteristics of these clusters?
- * *RQ3*. From all the questions that users are exposed to in the CSEd SE’s core network of interactions, what factors predict the questions that users to respond to?

Method

Data Collection

The data from the CSEd SE forum were collected through data mining, using the statistical software *R* (R Core Team, 2018). In particular, we used the *R* package *stackr* (Robinson, 2018) to extract data from the CSEd SE website. The *stackr* package helps obtain the read-only features of the Stack Exchange API with the ability to download information on questions, answers, users, tags, and other aspects of the site so that they can be analyzed in *R*.

Using *stackr*, we were able to collect the metadata for CSEd SE questions, answers, comments, tags, and users. The data are representative of all the information generated in the website from May 2017 (site launch) to June 2018. These metadata included information such as question identity number, answer identity number, user identity number, tag name, reputation scores of users, the date they joined the forum, and so on. This data collection resulted in a corpus of 559 questions asked by 210 different users, 2,675 answers contributed by 651 different users, and 7,209 comments from CSEd SE.

Data Analysis

First, we loaded the necessary *R* packages and input our data.

We then reshaped the data for analysis. We used the *R* package *igraph* (Csardi, 2018) to create a graph of the CSEd SE’s *core network of interactions*, focusing on nodes in the network (i.e., contributors) with degree of three or higher. In other words, we excluded users who participated as “tag-alongs,” perhaps answering a question or two but not interacting with other users in a significant way.

Results

RQ1. What are the characteristics of users who contributed to the CSEd SE’s core network of interactions?

To answer RQ1, we first tallied the number of questions, answers, and contributors to the CSEd SE during our data collection time period.

```
cse_d_qes$question_id %>% unique %>% length # number of distinct questions in CSEd SE
```

```
## [1] 559
```

```
cse_d_qes$display_name %>% unique %>% length # number of total, distinct contributors
```

```
## [1] 210
csed_ans$question_id %>% unique %>% length # number of questions answered in our time window

## [1] 553
csed_ans$answer_id %>% unique %>% length # number of answers in our time window

## [1] 2675
csed_ans$display_name %>% unique %>% length # number of people who answered questions in our time window

## [1] 651
## these are the different responders in the core network
csed_full_weighted %>% filter(weight > 0) %>%
  select(responder) %>% unlist %>% unique %>% length

## [1] 169
## these are the different question askers in the core network
csed_full_weighted %>% filter(weight > 0) %>%
  select(asker) %>% unlist %>% unique %>% length

## [1] 163
## these are total, distinct contributors (askers & responders) in the core network
c({csed_full_weighted %>% filter(weight > 0) %>% select(responder) %>% unlist},
  {csed_full_weighted %>% filter(weight > 0) %>% select(asker) %>% unlist}
) %>% unique %>% length
```

```
## [1] 257
```

We then calculated descriptive statistics.

```
##           Degree In-Degree Out-Degree
## mean:      9.59      4.79      4.79
## sd:        16.95      8.89     10.00
## median:     5.00      3.00      3.00
## min:        1.00      0.00      0.00
## max:       142.00     74.00     99.00
## skewness:   5.16      4.30      5.45
## kurtosis:  34.30     26.03     41.99
```

RQ2. Does network clustering occur in the context of the CSEd SE's core network of interactions? If so, what are some characteristics of these clusters?

As a reminder, a quick summary of the the CSEd SE core network of interactions during the time period of our study:

```
##
## Number of nodes: 257
## Number of edges: 1232
## Diameter: 6
```

Clustering: Spinglass Algorithm

A community is a set of nodes with many edges inside the community and few edges between outside it (i.e. between the community itself and the rest of the graph).

The *spinglass clustering algorithm* maps community detection onto finding the ground state of an infinite range spin glass. Csardi, Nepusz, and Airoldi (2016, pp. 132-133) explained:

The clustering method of Reichardt and Bornholdt (2006) is motivated by spin glass models from statistical physics. Such models are used to describe and explain magnetism at the microscopic scale at finite temperatures. Reichardt and Bornholdt (2006) drew an analogy between spin glass models and the problem of community detection on graphs and proposed an algorithm based on the simulated annealing of the spin glass model to obtain well-defined communities in a graph. A spin glass model consists of a set of particles called spins that are coupled by ferromagnetic or antiferromagnetic bonds. Each spin can be in one of k possible states. The well-known Potts model then defines the total energy of the spin glass with a given spin configuration. . . Spins and interactions in the Potts model are very similar to graphs: each spin in the model corresponds to a vertex, and each interaction corresponds to an edge. . . Reichardt and Bornholdt (2006) gave efficient update rules for the above energy function, making it possible to apply a simulated annealing procedure to find the ground state of the model that corresponds to a low energy configuration. Their algorithm starts from a random configuration of spins and tries to flip all the spins once in each time step. After each individual spin flip, the energy of the new configuration is evaluated.

In other words, the spinglass clustering algorithm partitions the vertices into communities by optimizing an energy function. The energy is optimized using the following function (Reichardt and Bornholdt, 2008):

$$H(\sigma) = - \sum (a_{ij} \text{internal links}) + \sum (b_{ij} \text{internal non-links}) + \sum (c_{ij} \text{external links}) - \sum (d_{ij} \text{external non-links})$$

This function penalizes (1) missing edges or non-links of people/nodes in the same cluster and (2) present links or edges between people/nodes in different clusters. It also rewards (1) present links or edges between people/nodes in the same cluster and (2) missing links or edges between people/nodes in different clusters. Thus, a lower score (i.e., energy level) is better as it means that the internal links and external non-links have more weightage in that model. In other words, in a strong model, members within clusters are strongly linked and members in separate clusters are weakly linked. Here, $a_{ij}, b_{ij}, c_{ij}, d_{ij}$ represent the individual weights of the four components.

The initial R code to produce spinglass clusters is straightforward:

```
csg_0 <- csed_graph %>% cluster_spinglass # creates the clusters
csg_0$membership %>% unique %>% length # number of clusters/communities/groups
```

```
## [1] 7
```

One of the important outcomes of this method is the *modularity* value M . Modularity measures how good the division is, or how separated are the different vertex types from each other. The spinglass algorithm looks for the modularity of the optimal partition. For a given network, the partition with maximum modularity corresponds to the optimal community structure (i.e., a higher M is better).

Note also that if $M = 0$, all nodes belong to one group; if $M < 0$, each node belongs to separate community.

```
csg_0$modularity # The modularity of a graph with respect to some division (or vertex types)
```

```
## [1] 0.2773503
```

Identifying the “Typical” Number of Clusters Returned with the Spinglass Algorithm

It is important to note that a different result is returned each time the spinglass clustering algorithm is run. For this reason, we needed to run a number of simulations to see what the “typical” number of clusters are.

We ran the algorithm 100 times and looked at the mean and median number of clusters obtained. We made a note of a *seed* that produced the median number of clusters, confirmed that this was reproducible, and then set this seed so that all future work will be run with this same clustering configuration.

```
##
## number of tests: 100.00
## mean:          7.10
## sd:            0.86
## min:           6.00
## max:           10.00
## skewness:      0.48
## kurtosis:      3.11
## median:        7.00

##
## Number of nodes:      257.0000
## Number of edges:      1232.0000
## Number of clusters:    7.0000
## Modularity:           0.2786

## [1] Size of each cluster:
## [1] 36 31 36 48 43  4 59
```

Test of Statistical Significance for Spinglass Clusters

The test for statistical significance for spinglass clustering is a bit different than the familiar tests that return *p*-values (Csardi, Nepusz, & Airoldi (2016, pp. 132-138).

The idea behind this test of significance is that a random network of equal size and degree distribution as our observed network should have a lower modularity score—that is, if the observed network does in fact have statistically significant clustering.

The following R procedure generates 100 randomized instances of our network (with the same size and degree distribution) using the `sample_degseq()` function. The `method = 'v1'` ensures that there are no loop edges in the randomly generated networks. We then applied the spinglass clustering algorithm to each of the 100 randomized instances of the network.

A '0' result from this procedure indicates that no randomized networks have community structure with a modularity score that is higher than the one obtained from the original, observed network. Hence a '0' result means that our network has significant community structure; any non-zero results means that the detected spinglass clusters are not statistically significant.

```
#degrees <- csed_graph %>% as.undirected %>% degree(mode='all', loops=FALSE)
#qr_vl <- replicate(100, sample_degseq(degrees, method="v1"),
#                    simplify=FALSE) %>%
#    lapply(cluster_spinglass) %>%
#    sapply(modularity)
#saveRDS(qr_vl, "model-output/qr_vl.rds")
qr_vl <- read_rds("model-output/qr_vl.rds")
sum(qr_vl > csg$modularity) / 100
```

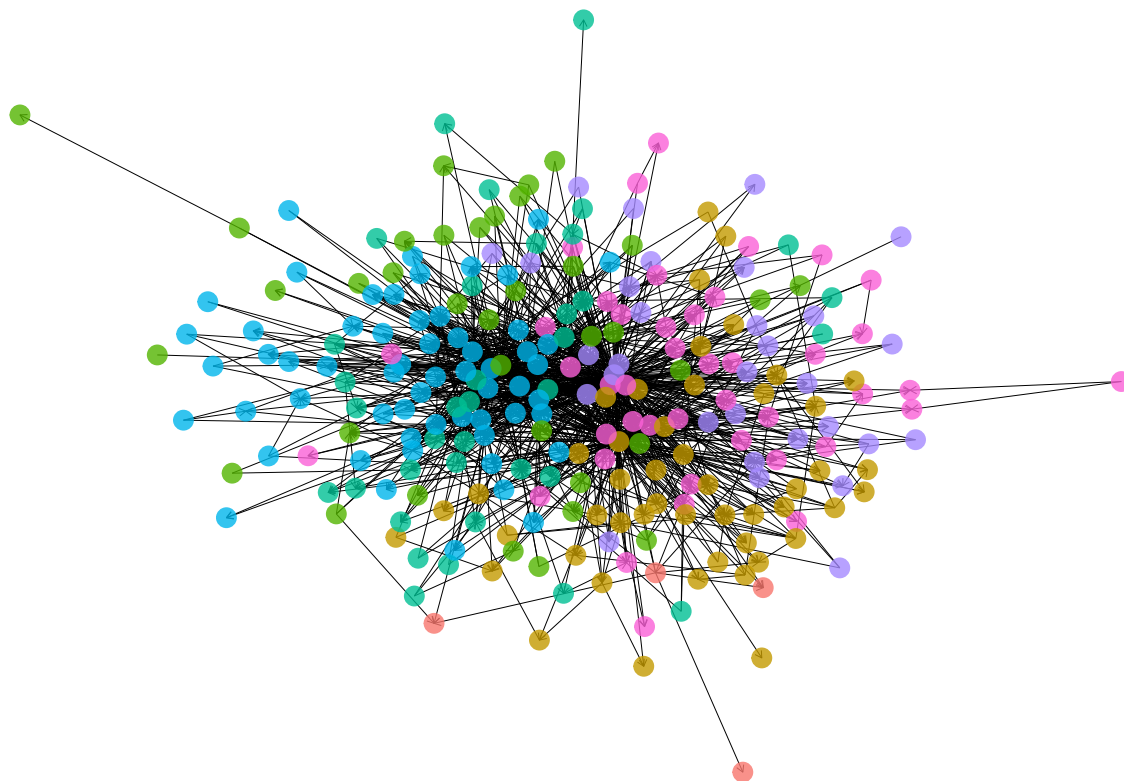
```
## [1] 0
```

Network Visualization with Clusters

Finally, we created a visualization of our network structure, using the color palette generated by our spinglass clustering. Here, we used the *Fruchterman-Reingold layout algorithm* (`layout = 'fr'`), which is appropriate

for large (but still with less than 1,000 nodes), potentially disconnected networks.

```
## IGRAPH 1de4529 DN-- 257 1232 --
## + attr: name (v/c), degree (v/n), in_degree (v/n), out_degree
## | (v/n), cluster_weight (e/n)
```



RQ3. What factors predict the questions that users to respond to?

Conceptualizing the Selection Model

We chose to use a p_2 model for conceptualizing the selection process. We did this so that we could account for fixed effects of dyads, receivers (i.e., question askers), and senders (i.e., question responders) in addition to random effects of receivers and senders.

Here are the levels of the model:

Null p_2 Selection Model: Responder and Asker Random-Effects

We started by building the null p_2 model for predicting ties (i.e., answering someone else's question on the CSEd SE). This model only considered responder and asker random effects, which would include in-degree and out-degree.

Pair Level p_2 Selection Model

Our pair level p_2 selection model for predicting ties added membership in the same clusters as a fixed effect in addition to the responder and asker random effects from the null p_2 .

This model considers sender and receiver fixed effects along with any random effects due to senders and receivers. The selection model for responder i answering a question asked by user j :

$$\log\left(\frac{p(w_{ii'})}{1 - p(w_{ii'})}\right) = \theta_{00} + \theta_{0i} + \theta_{0i'} + \theta_{0q} + \theta_1|x_i - x_{i'}| + \theta_2|y_i - y_{i'}| + \theta_3|a_{iqt0} - a_{iqt1}| + \rho(w)_{ii'} + \epsilon_0$$

Where:

1. θ_{00} = Intercept
2. θ_{0i} = Sender/reponder effects
3. $\theta_{0i'}$ = Receiver/asker effects
4. θ_{0q} = Question effects
5. $|x_i - x_{i'}|$ = Difference in profession
6. $|y_i - y_{i'}|$ = Difference in gender
7. $|a_{iqt0} - a_{iqt1}|$ = Difference in answering in topic area at times t0 and t1. That is, prior ties/relationship of user i to a topic area
8. $\rho(w)_{ii'}$ = Reciprocity of i and i'
9. w = Answering a certain question

Sender Level p_2 Selection Model for User i Responding to the Question:

Our sender level p_2 selection model for predicting ties added responder fixed effects related to reputation, role, time on the platform, and questions asked in addition to the responder random effects.

$$\theta_{0i} = \alpha_0 + \alpha_1(\text{Reputation points}_i) + \alpha_2(\text{Editor/Moderator}_i) + \alpha_3(\text{Duration of membership}_i) + \alpha_4(\text{Number of questions previously answered}_i) + \epsilon_1$$

Where

1. Reputation points = The numerical reputation points associated with each user.
2. Editor/Moderator = 1 if the responder answering question is an editor or a moderator. 0 otherwise.
3. Duration of Membership = Time in months of being a member of CSEd SE
4. Number of questions previously asked = Total number of questions asked up until present time.

Receiver Level p_2 Selection Model for User i' Asking the Question:

$$\theta_{0i'} = \beta_0 + \beta_1(\text{Reputation points}_{i'}) + \beta_2(\text{Duration of membership}_{i'}) + \beta_3(\text{Number of questions previously asked}_{i'}) + \epsilon_2$$

Where

1. Reputation points = The numerical reputation points associated with each user.
2. Duration of Membership = Time in months of being a member of CSEd SE.
3. Number of questions previously asked = Total number of questions asked up until present time.¹

p_2 Selection Model for Question q :

$$\theta_{0q} = \gamma_0 + \gamma_1(\text{Number of votes}_q) + \gamma_2(\text{Number of answers}_q) + \gamma_3(\text{Number of views}_q) + \epsilon_3$$

Where

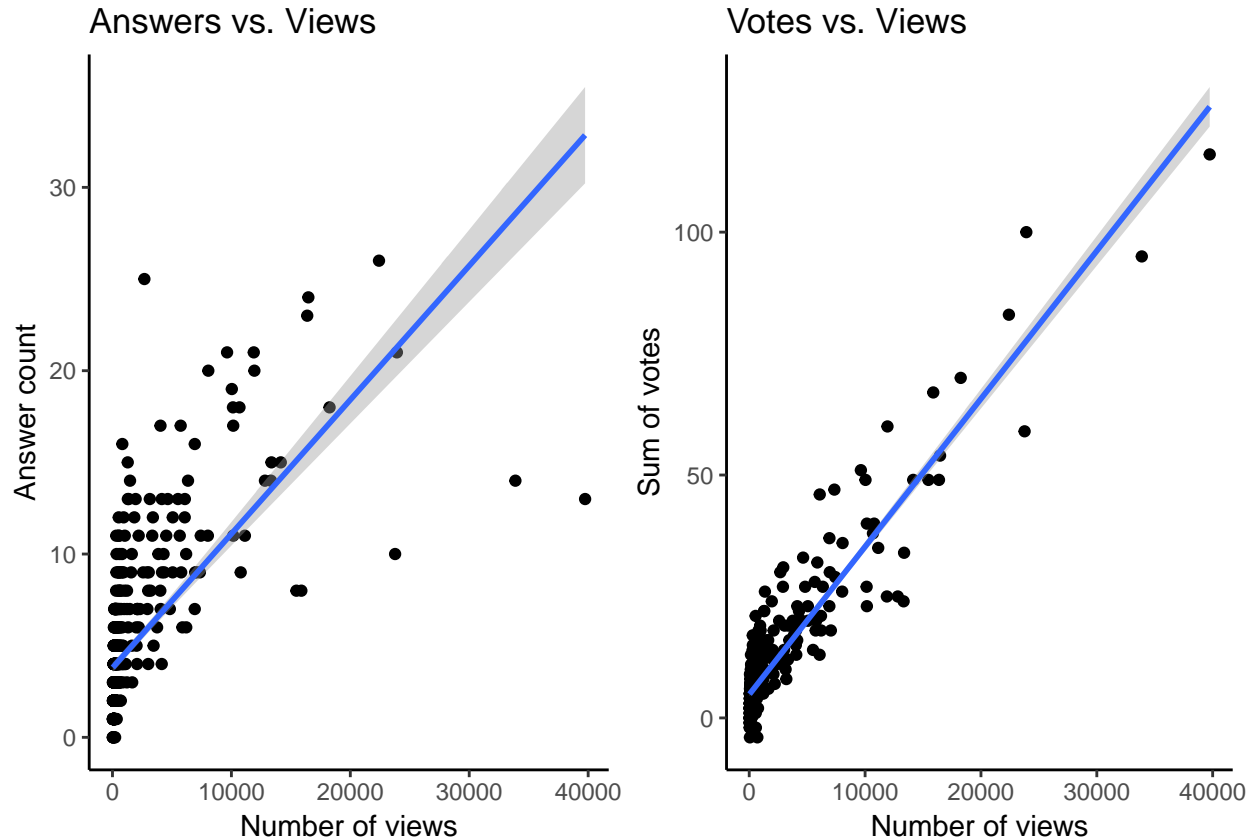
1. Number of votes = Total up-votes the question got on the forum prior to i answering the question.
2. Number of answers = Total answers the question had prior to i answering the question.
3. Number of views = Total views the question got on the forum prior to i answering the question.

¹We did not add Editor/Moderator role for askers, as when users are responding to an asker, they will not immediately know if the asker is an editor or moderator. The responder will only know this if they click and read the asker's profile. Given this, we decided to not include it in the model.

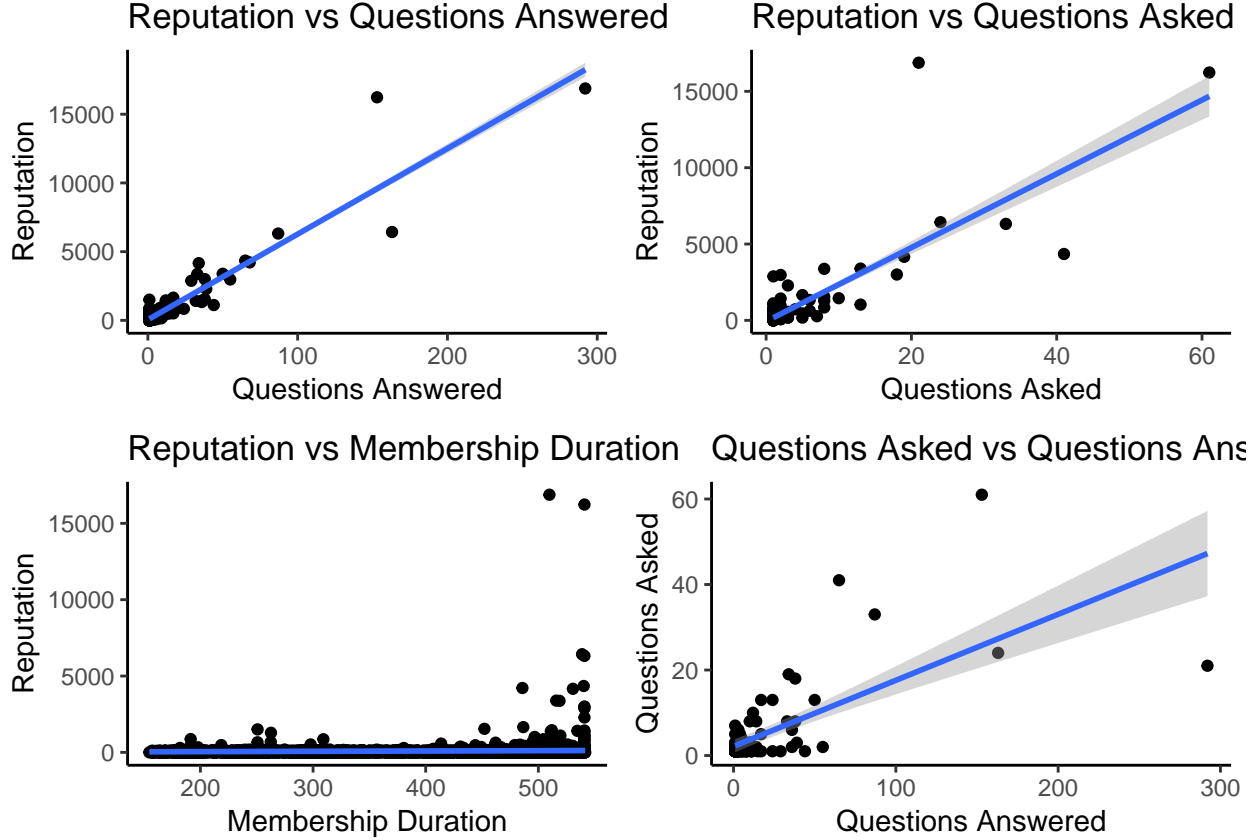
Identifying Sources of Multicollinearity

We tested to see if these variables are highly correlated and if we needed to change the models that we proposed. These are the correlations that we tested:

1. Correlations between number of votes, answers, and views
2. Correlation between reputation points and number of questions answered/asked
3. Correlation between reputation points and membership duration
4. Correlation between questions asked and answered



```
## [1] Correlations:
##           view_count answer_count score
## view_count      1.00      0.67  0.92
## answer_count    0.67      1.00  0.74
## score           0.92      0.74  1.00
## Warning: Removed 4445 rows containing non-finite values (stat_smooth).
## Warning: Removed 4445 rows containing missing values (geom_point).
## Warning: Removed 4879 rows containing non-finite values (stat_smooth).
## Warning: Removed 4879 rows containing missing values (geom_point).
## Warning: Removed 8 rows containing non-finite values (stat_smooth).
## Warning: Removed 8 rows containing missing values (geom_point).
## Warning: Removed 5000 rows containing non-finite values (stat_smooth).
## Warning: Removed 5000 rows containing missing values (geom_point).
```



[1] Correlations:

```
##               reputation questions_answered questions_asked
## reputation               1.00              0.94             0.81
## questions_answered       0.94              1.00             0.68
## questions_asked          0.81              0.68             1.00
## membership_duration_days 0.18              0.18             0.17
##
##               membership_duration_days
## reputation                   0.18
## questions_answered           0.18
## questions_asked              0.17
## membership_duration_days     1.00
```

Updated Models

New Sender Level p_2 Selection Model for User i Responding to the Question:

$$\theta_{0i} = \alpha_0 + \alpha_1(\text{Reputation points}_i) + \alpha_2(\text{Editor/Moderator}_i) + \alpha_3(\text{Duration of membership}_i) + \Theta_{zi} + \epsilon_1$$

(We removed 'number of questions previously answered' as it was highly correlated with reputation points. We added random effects, Θ_{zi} in case that plays a role in receiver/asker effects)

New Receiver Level p_2 Selection Model for User i Asking the Question:

$$\theta_{0i'} = \beta_0 + \beta_1(\text{Reputation points}_{i'}) + \beta_2(\text{Duration of membership}_{i'}) + \Theta_{zi'} + \epsilon_2$$

(We removed ‘number of questions previously asked’ as it was highly correlated with reputation points. We added random effects, Θ_{zi} , in case that plays a role in receiver/asker effects)

New p_2 Selection Model for Question q :

$$\theta_{0q} = \gamma_0 + \gamma_1(\text{Number of votes}_q) + \Theta_{zq} + \epsilon_3$$

(We removed ‘number of answers’ and ‘number of views’ as they were highly correlated with number of votes)

The variables in the current model reflect prior research on teacher interactions where things such as ‘reputation’, years of service (in CSEd SE it is ‘membership duration’, formally designated leaders (in CSEd SE it is ‘editor/moderator’), etc mattered for the selection model.

Constructing the Selection Models Using Linear Mixed-Effects Models

Having conceptually described the selection models of users on CSEd SE choosing to answer certain questions, we then built these in R using *generalized linear mixed-effects* (GLME) with the *lme4* package (Bates, Maechler, Bolker, & Walker, 2018). We tried to fit these GLME models by incrementally adding factors; in total we created five models: (a) the null p_* with only responder and asker random effects, (b) model 1 which added responder and asker reputation scores as fixed effects, (c) model 2 which added responder and asker roles as fixed effects, (d) model 3 which added responder and asker time on the CSEd SE as fixed effects, and (e) model 4 which added responders’ questions answered and askers’ questions asked as fixed effects.

For each model, we calculated the *standard error of the mean* (SEM) of the conditional modes of the random-effects coefficients from the fitted model; SEM equals the standard deviation divided by the square root of the sample size. We also used the R package *sjstats* (Ludecke, 2018b) to report the *intraclass-correlation coefficient* (ICC) scores for responders and askers; the ICC represents the percent of variability due to the group. Finally, we compared the *Akaike information criterion* (AIC) and the *Bayesian information criterion* (BIC), keeping in mind that smaller values are better.

We present SEM, ICC, AIC, and BIC scores for all five p_2 selection models in a summary table at the end. Finally, for each selection model, we used the *sjPlot* R package (Ludecke, 2018a) to create plots showing the conditional modes of the random-effects coefficients and standard errors by contributor, in other words, the difference between the average predicted response for a given set of fixed-effect values (treatment) and the response predicted for a particular individual. This is a method to see how much any individual differs from the population. We examined these plots to see how they changed as we added factors to the selection model.

Null p_2 Selection Model: Responder and Asker Random-Effects

We started by building the null p_2 model for predicting ties (i.e., answering someone else’s question on the CSEd SE).

Pair Level p_2 Selection Model

With our current level of analysis, we are unable to construct the pair-level p_2 selection model at this time.

Sender Level p_2 Selection Model

We constructed a sender level p_2 selection model for predicting ties by incrementally adding responder fixed effects related to (a) reputation, (b) role, (c) time on the platform, and (d) questions answered in addition to the responder and asker random effects. However, the third model failed to converge and so we have not included the responders’ time on the CSEd SE in the results.

p_2 Sender-Level Summary Table

Finally, we compiled the statistics from the null p_2 model and the three incremental stages of the sender level p_2 model (i.e., the ones that converged) and compiled them in a table for comparison (see below). We noted that the standard errors, AIC, and BIC all generally decreased as we iteratively added features to the models.

##		Null Reputation		Role Questions	
## Responder	SE	1.3113	0.0388	0.0376	0.0000
## Responder	ICC	0.3171	0.0653	0.0638	0.5055
## Asker	SE	0.9305	0.9612	0.9579	0.9929
## Asker	ICC	0.3481	0.4768	0.4772	0.0000
## AIC		8245.2017	7989.7014	7988.0470	7555.2270
## BIC		8272.4961	8026.0940	8042.6359	7616.2639

Receiver Level p_2 Selection Model

We constructed a receiver level p_2 selection model for predicting ties by incrementally adding asker fixed effects related to (a) reputation, (b) time on the platform, and (c) questions answered in addition to the responder and asker random effects. However, the second model failed to converge and so we have not included the askers' time on the CSEd SE in the results.

p_2 Receiver-Level Summary Table

Finally, we compiled the statistics from the null p_2 model and the two incremental stages of the sender level p_2 model (i.e., the ones that converged) and compiled them in a table for comparison (see below). We noted that the standard errors, AIC, and BIC all generally decreased as we iteratively added features to the models.

##		Null Reputation		Questions	
## Responder	SE	1.3113	1.3113	1.3745	
## Responder	ICC	0.3171	0.3338	0.4720	
## Asker	SE	0.9305	0.3015	0.0261	
## Asker	ICC	0.3481	0.3139	0.0270	
## AIC		8245.2017	8235.4024	7734.7783	
## BIC		8272.4961	8271.7950	7778.0836	

Discussion

The answers to our first research question revealed that there is a wide range in user characteristics of the members who participated in the 'core interactions' in CSEd SE. The highest 'degree' a user had was 142. The lowest was 1. The median (5) and mean (10) were low as well. This reveals that in the forum, there are only a few users who contribute to the most number of interactions of asking and answering questions. This trend is seen in many online discussion forums and social media sites. A comparison of these websites can possibly reveal which forums have a wider level (more users contributing to conversations as opposed to a few users contributing to most conversations) of interactions. Practitioners might find such information useful if directing teachers to various forums as a part of their professional development. It can also be useful from a 'forum development' point of view if certain forum characteristics are correlated with wider user interactions.

The answer to the second research question revealed that there indeed is clustering that happens in the forum. A content analysis of the user profiles and question characteristics will be helpful to find out the reasons for clustering. This information might be useful for teachers participating in these forums, in case they are in search for a specific group to interact with or learn from, be in terms of geographic location or topic area.

The answers to the third research question will reveal the reasons for users to answer questions in the forum. In the past, only high reputation points were seen predictive of users answering more questions. It will be interesting to find what other characteristics influence someone to answer questions in a forum. Further, 'lurking' is a common phenomenon that happens in online contexts and little is understood why people choose to not participate/answer questions. By finding why people *do* participate and answer questions, we might be able to move a step towards understanding why people *do not* participate or answer questions.

Future Research

1. We anticipate adding variables to the selection model, such as users' geographical location, gender, and profession. With these additional variables, we should be able to construct a pair-level p_2 . In order to find the gender and profession of individuals, we will have to conduct content analysis of the members in the forum. Not all members specify these traits, so we might have to conduct a survey in the forum.
2. We will use KliqueFinder to cross-check the spinglass clusters derived in this current work. One of the biggest differences between the spinglass and Kliquefinder approaches is that spinglass uses addition to optimize for links/non-links between and within clusters, as mentioned earlier; Kliquefinder, on the other hand, uses a slightly different method for optimization, which we believe could be better suited for small networks.
3. We will find the focii of the clusters so that we can find out if the clusters are centered around users with high social capital, around certain topic areas, geography, or profession.
4. With additional clarity around clusters, we will add aesthetics to the network visualization, such as making the node size reflect the $\log()$ of a user's reputation score.
5. Finally, we will also use content analysis to explore the user characteristics of those who have a high in-degree and out-degree.

References

- Bates, D., Maechler, M., Bolker, B., & Walker, S. (2018). lme4: Linear mixed-effects models using 'Eigen' and S4 (Version 1.1-19) [R package]. Retrieved from <https://cran.r-project.org/package=lme4>
- Brown, N. C. C., & Kolling, M. (2013, August). A tale of three sites: Resource and knowledge sharing amongst computer science educators. In *Proceedings of the ninth annual international ACM conference on International computing education research* (pp. 27-34). ACM.
- Csardi, G. (2018). igraph: Network analysis and visualization (Version 1.2.2) [R package]. Retrieved from <https://CRAN.R-project.org/package=igraph>
- Ludecke, D. (2018). sjPlot: Data visualization for statistics in social science (Version 2.6.1) [R package]. Retrieved from <https://cran.r-project.org/package=sjPlot>
- Ludecke, D. (2018). sjstats: Collection of convenient functions for common statistical computations (Version 0.17.2) [R package]. Retrieved from <https://cran.r-project.org/package=sjstats>
- Macia, M., & Garcia, I. (2016). Informal online communities and networks as a source of teacher professional development: A review. *Teaching and Teacher Education*, 55, 291-307.
- Movshovitz-Attias, D., Movshovitz-Attias, Y., Steenkiste, P., & Faloutsos, C. (2013, August). Analysis of the reputation system and user contributions on a question answering website: Stackoverflow. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 886-893). ACM.
- Ni, L., Guzdial, M., Tew, A. E., Morrison, B., & Galanos, R. (2011, March). Building a community to support HS CS teachers: the disciplinary commons for computing educators. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 553-558). ACM.
- Pedersen, T. L. (2018). ggraph: An implementation of grammar of graphics for graphs and networks (Version 1.0.2) [R package]. Retrieved from <https://CRAN.R-project.org/package=ggraph>
- R Core Team. (2018). R: A language and environment for statistical computing (Version 3.5.0) [Computer software]. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>

- Reichardt, J., & Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74(1), 016110.
- Robinson, D. (2015). stackr: An R package for connecting to the Stack Exchange API [R package]. Retrieved from <https://github.com/dgrtwo/stackr>
- Trust, T., Krutka, D. G., & Carpenter, J. P. (2016). ???Together we are better???: Professional learning networks for teachers. *Computers & education*, 102, 15-34.
- Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. New York, NY: Cambridge University Press.
- Wenger, E. (2011). *Communities of practice: A brief introduction*.
- Wickham, H., Chang, W., & RStudio. (2016). ggplot2: Create elegant data visualisations using the grammar of graphics (Version 2.2.1) [R package]. Retrieved from <https://CRAN.R-project.org/package=ggplot2>
- Wickham, H., Francois, R., Henry, L., Muller, K., & RStudio. (2018). dplyr: A grammar of data manipulation (Version 0.7.6) [R package]. Retrieved from <https://CRAN.R-project.org/package=dplyr>
- Yadav, A., Gretter, S., Hambruch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education*, 26(4), 235-254.