2013 IEEE 13th International Conference on Data Mining

Mining User Lifecycles from Online Community Platforms and their Application to Churn Prediction

Matthew Rowe
School of Computing and Communications
Lancaster University
Lancaster, UK
Email: m.rowe@lancaster.ac.uk

***Abstract*—Recent work has studied user development in the domains of both telecommunication and online community platforms, examining how users develop in terms of the company they keep (socially) [1] and the language they use (lexically) [2]. Such works afford key insights into user changes along individual dimensions, yet they do not examine how users develop relative to their prior behaviour along multiple dimensions. In this paper we examine how users develop along various properties (in-degree, out-degree, posted terms) in three online community platforms (Facebook, SAP Community Network, and Server Fault) and using three models of user development: (i) isolated lifecycle periods, (ii) historical contrasts, and (iii) community contrasts. We present an approach to mine the lifecycle trajectories of users as a means to characterise user development along the different properties and development models, and demonstrate the utility of such trajectories in predicting churners. We find consistent effects with past work: users tend to reflect the behaviour of the community in early portions of their lifecycles, before then diverging from the community towards the end. We also find that users form sub-communities with whom they communicate and remain within.**

***Keywords*-online communities, social networks, user development, user lifecycles, churn prediction**

I. Introduction

How people develop over the course of their lives has been the subject of investigation in various domains (psychology, psychoanalysis, anthropology, etc.). Early work by the psychoanalyst analyst Erik H. Erikson [3] examined the development that individuals undertake in cultivating their own identity, he theorised that people are susceptible to alterations within the adolescent stages of their lives in which they often experience an *identity crisis*. Erikson outlined a series of development periods that individuals transcend: childhood curiosity, adolescent identity confusion, connecting with others in early adulthood, etc.

Recent work has advanced the examination of such development stages into the domains of both telecommunications [1], in order to predict users leaving telecommunication providers, and online community platforms [2], [4]. Such works have thus far concentrated on isolated user properties, i.e. either social or lexical dynamics, in order to understand how users develop socially over time [1], or lexically (in terms the language they use) relative to the community in which they are interacting [2]. Therefore at present there is no understanding of how users develop throughout their *lifecycles* in online community platforms both socially (in terms of the company they keep) and lexically (in terms of the language they use), nor how users develop in comparison to their previous social or lexical characteristics. In understanding the lifecycles that users go through and the key features that discern different lifecycle periods we can: (i) recommend items to users by building a neighbourhood from users within the same development stage; and (ii) predict who is likely to churn from the online community.

In this paper we present work in the direction of modelling user lifecycles in online community platforms using both social and lexical properties, and the development of users over time relative to their past behaviour and the community in which they are interacting. Our contributions are four-fold:

1) We model users based on their social and lexical dynamics, thus complementing recent work [1], [2]
2) We assess the lifecycle development of users using three online community platform datasets (Facebook, SAP Community Network and ServerFault): (i) over individual lifecycle periods, (ii) relative to their past lifecycle periods, and (iii) relative to the community in which they are interacting.
3) We mine user lifecycle trajectories using different user properties and development indicators, and characterise the development of users as a result.
4) We apply the mined trajectories to the task of churn prediction and demonstrate their utility through significantly outperforming random model baselines.

We begin the paper by explaining the related work within the fields of social network evolution and lifecycle mining.

II. Related Work

Examining and modelling the changes that users go through and undertake in online social networks and communities has been explored in a variety of works. Leskovec et al. [5] assessed the evolution of social network structures on

1550-4786/13 $31.00 © 2013 IEEE
DOI 10.1109/ICDM.2013.78

Flickr, Delicious, Answers and LinkedIn, using maximum-likelihood estimation to fit models of node arrival and edge creation rates. The authors showed similar effects in terms of the social processes at work throughout the social platforms. Panzarasa et al. [6] used an online community platform provided for students of the University of California, Irvine, to assess how the social network structure of the community platform evolved over time. The authors found that certain users to acted consistently as hubs through which communication was mediated. Similar work by Backstrom et al. [7] explored the effects that govern group formation and joining behaviour on LiveJournal, and found that using the proportion of a user's friends already within a group had a key effect on identifying group joiners. Kairam et al. [8] assessed the dynamics of group formations within the social networking platform Ning. Like Backstrom et al., the authors found that the probability of a user joining a group was linked to the number of prior members within whom he has a relationship. Recent work by Gong et al. [9] inspected the evolution of social networks on Google+ as the platform was growing in memberships, in particular they focused on *social-attribute networks* (i.e. bipartite graphs containing people and their attributes as nodes), finding that the platform exhibited unique growth and characteristics of the networks as more people joined Google+ (i.e. reduced reciprocity). Chung et al. [10] examined the assortativity (i.e. the degree to which similar degree nodes interact with one another) of social networks derived from an online community building web site over a ten-year period, and found assortativity to increase with time, while the community platform remained largely dissasortive.

The aforementioned works provide interesting insights into the evolution of social networks and groups in online communities and serve to unearth the processes that underly such evolution. However they are limited in modelling the lifecycles that users exhibit and the extent to which users differ in how they they evolve over time. Recently, several works have attempted to characterise the individuality of users in terms of their evolution throughout their lifecycles within social platforms. For instance, Miritello et al. [1] examined the social evolution of users over time in terms of: a) communication capacity of users (the limit of the number of social connections they can maintain), and b) the activity rate of users (the number of edges users created), using call records data. The authors found that as people aged throughout their lifecycle that their social circle reduced in size and that interaction occurs less towards later life periods. Similarly Danescu et al. [2] assessed the lexical dynamics of online community members by modelling their term distributions and how these changed relative to the community. The authors examined two beer-rating communities (Beer Advocate and Rate Beer) and found that users began their lifecycle within the community by adapting their language to the community but then stopped doing so. McAuley and Leskovec [4] examined how users evolved in their expertise (assuming a monotonic progression) over time in the same beer rating communities as [2]. The authors defined users as evolving based on their own '*personal clock*' where the rate of progression is user-dependent, and used this notion to mine latent experience levels for each user and then use these for recommendations.

In this paper we complement the work of Miritello et al. [1] and Danescu et al. [2] by examining both social and lexical dynamics of users in terms of their evolution. We compare such dynamics not only with the community (as in [2]) but also relative to earlier time periods to understand how users evolve in relation to their past properties. In the following section we explain the datasets used for our experiments and how we define user lifecycle periods, using the same notion of a *personal clock* from [4], before going on to explain the characterisation of user properties and their evolution throughout lifecycle periods.

Table I
STATISTICS OF THE ONLINE COMMUNITY PLATFORM DATASETS.

Platform	Time Span	Post Count	User Count
Facebook	[18-08-2007,24-01-2013]	118,432	4,745
SAP	[15-12-2003,20-07-2011]	427,221	32,926
Server Fault	[01-08-2008,31-03-2011]	234,790	33,285

III. DATASETS

To provide a broad examination of user lifecycles across different online community platforms we used data collected from three independent sources: Facebook, the SAP Community Network (SAP) and Server Fault. We examined all users who have posted more than 40 times throughout their lifecycle.[1] Table I provides summary statistics of the datasets.

Facebook data was obtained from Facebook groups related to Open University degree course discussions. The groups allow users to discuss the problems and issues that they may be having with degree course material and potential avenues for solving those problems. Although Facebook provides the ability to collect social network data for users, we did not collect such data in this instance and instead used the reply-to graph within the groups to build social networks for individual users. In doing so we would constrain the social dynamics at play to those within the context of the groups.

The SAP Community Network is a community question answering system related to SAP technology products and information technologies. Users sign up to the platform and post questions related to technical issues, other users then provide answers to those questions and should any answers satisfy the original query, and therefore solve the issue, the answerer is awarded points. To construct social

[1]Choosing 40 so that we have at least 2 posts per lifecycle period.

networks, and hence derive our social features that we will define below, we use the reply-to graph to form implicit connections between users. Similar to SAP, Server Fault is a platform that is part of the Stack Overflow question answering site collection.[2] The platform functions in a similar vein to SAP by providing users with the means to post questions pertaining to a variety of server-related issues, and allowing other community members to reply with potential answers. Similar to SAP, Server Fault also lacks explicit edge-creation features, therefore we use the reply-to graph (i.e. where a user has replied to another user's question) to form an implicit edge between the users.

IV. Modelling User Lifecycles

To examine how users develop and evolve throughout their lifecycles we first needed to define a means to characterise a user by their lifecycle and thus examine individual stages in terms of changes in user properties. Before defining such lifecycles, and their characterisation, we divided users from each dataset up into an 80/20% random split and used the former set for training and analysis, as described below, and the latter split for churn prediction experiments that we present later in the paper.

A. Defining Lifecycle Periods

Existing recent work [1], [2], [4] has demonstrated the extent to which users develop at their own pace and thus evolve according to their own '*personal clock*' [4]. We validated this finding in the context of our datasets by deriving each user's lifetime in the system into 20-equally time sliced windows and examining the proportion of the user's activity (posts) within each windowed interval. We found that for each platform user activity peaks at the start of their lifetimes, before reducing and then increasing again towards the end, suggesting that users join the community and participate initially, before reducing their activity over time gradually.

Algorithm 1 Deriving the set of lifecycle periods (T)

$chunkSize \leftarrow |P_{u_i}|/20$
$Q_{u_i} \leftarrow sort(P_{u_i})$
$i \leftarrow 0$
$T \leftarrow \emptyset$
while $i < 20$ **do**
  $start \leftarrow i \times chunkSize$
  $end \leftarrow (i+1) \times chunkSize$
  if $end > |Q_{u_i}| - 1$ **then**
    $end = |Q_{u_i}| - 1$
  end if
  $t_i \leftarrow time(Q_{u_i}[start])$
  $t_j \leftarrow time(Q_{u_i}[end])$
  $T \leftarrow T \cup \{[t_i, t_j]\}$
end while
return T

These findings suggest that a time-sliced approach to deriving the lifetime periods of a user would be inappropriate as the lack of activity within certain periods would have a strong effect on the social and lexical dynamics that could be observed. Therefore for deriving the lifecycle periods of users within the platforms we adopted an activity-slicing approach that divides a user's lifetime into 20 discrete time intervals but with an equal proportion of activity within each period; this approach is analogous to those adopted in prior work [1], [2], [4]. We defined this approach in Algorithm 1 which functions as follows: we derive the set of interval tuples ($\{[t_i, t_j]\} \in T$) by first deriving the chunk size (i.e. the number of posts in a single period) for each user, we then sort the posts in ascending date order, before deriving the start and end points of each interval in an incremental manner. This derives the set of time intervals T that are specific to a given user, these are then used to assess the evolution of users across disparate properties.

B. Modelling User Properties

Having defined the lifecycle periods we now move on to modelling user properties within discrete time intervals. In doing so we can track how the properties of users change over time, and model the trajectories that such changes make. We model user properties in terms of: (i) social dynamics, and (ii) lexical dynamics, thereby capturing how the social connectivity of users changes over time and how the language that users adopt changes.

1) In-degree and Out-degree Distributions: Our first user properties involve examining the *social dynamics* of each user in terms of his *in-degree* and *out-degree*. The former describes the number of edges that connect to a given user, while the latter describes the number of edges from the user. As we are dealing with conversation-based platforms for our experiments we can use the *reply-to* graph to construct these edges, where we define an edge connecting to a given user u_i if another user u_j has replied to him. Likewise, we also define an edge from a given user u_i to another user u_j if the former has replied to the latter.

Given our use of lifecycle periods we use the discrete time intervals that constitute $[t, t'] \in T$ to derive the set of users who replied to u_i, defining this set as $\Gamma^{IN}_{[t,t']}$. We also define the set of users that u_i that has replied to within a given time interval: $\Gamma^{OUT}_{[t,t']}$. From these definitions we can then form a discrete probability distribution that captures the distribution of repliers to user u_i, using $\Gamma^{IN}_{[t_i,t_j]}$, and user u_i responding to community users, and hence using $\Gamma^{OUT}_{[t_i,t_j]}$. For an arbitrary user ($u_j \in \Gamma^{IN}_{[t_i,t_j]}$) who has contacted user u_i within time segment $[t_i, t_j]$ we define this probability of

[2]http://stackoverflow.com/

interaction as follows:[3]

$$Pr(u_j \mid \Gamma^{IN}_{[t,t']}) = \frac{|\{q : p \in P_{u_i}, q \in P_{u_j}, t \leq time(q) < t', q \rightarrow p\}|}{\sum_{u_k \in \Gamma^{IN}_{[t,t']}} |\{q : p \in P_{u_i}, q \in P_{u_k}, t \leq time(q) < t', q \rightarrow p\}|} \quad (1)$$

And for an arbitrary user ($u_j \in \Gamma^{out}_{u_i t}$) who user u_i has contacted within time segment t we define the probability of interaction as follows:

$$Pr(u_j \mid \Gamma^{OUT}_{[t,t']}) = \frac{|\{p : p \in P_{u_i}, q \in P_{u_j}, t \leq time(p) < t', p \rightarrow q\}|}{\sum_{u_k \in \Gamma^{OUT}_{[t,t']}} |\{p : p \in P_{u_i}, q \in P_{u_k}, t \leq time(p) < t', p \rightarrow q\}|} \quad (2)$$

Given this formulation we now have time-dependent discrete probability distributions for a given user's in-degree and out-degree distribution, thereby allowing the *social* changes of users to be analysed in terms of the users communicating with a given user over time.

2) Term Distribution: We model the *lexical dynamics* of users based on their term usage over time. To derive the set of terms we first retrieve all posts made by a given user within a time interval and then remove stop words and filter out any punctuation. Having derived the set of cleaned posts, we then define the discrete probability distribution for a user u_i within interval $[t, t']$ based on the conditional probability of term x being used within the time interval. We define a multiset as containing the set of terms used by u_i in a given time period: $x \in C_{[t,t']}$ and a mapping function $\mu : C_{[t,t']} \rightarrow \mathbb{N}$ that returns the multiplicity of a given term's usage by the user at a given time period. Thus we define the conditional probability for term x being used by u_i during $[t, t']$ as:

$$Pr(x \mid [t, t']) = \frac{\mu(x)}{\sum_{x' \in C_{u_i t}} \mu(x')} \quad (3)$$

C. Analysing User Lifecycles

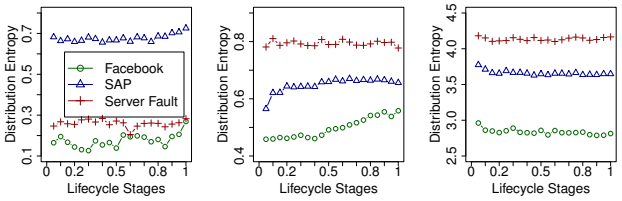
To examine how users' properties evolve throughout their lifecycles we inspected distribution changes using three means: (i) analysing the variation within the distribution in each lifecycle period; (ii) analysing the minimal change in variation when comparing one lifecycle period with earlier periods, thus informing how the user is changing relative to past behaviour; and (iii) analysing the variation in one lifecycle period relative to how the platform varies over the same time interval.

1) Inspecting Individual Periods (Period Entropy): To analyse the variation in a user's properties within a given lifecycle period we derived the entropy of each probability distribution within each lifecycle period. Entropy describes the amount of variation within a random variable, and therefore provides a useful means to gauge how much a given user is varying: (i) the people with whom he is communicating, and (ii) the terms that he is using within his posts. We define the entropy of an arbitrary probability distribution P as follows:

$$H(P) = -\sum_x p(x) \log p(x) \quad (4)$$

For each platform (Facebook, SAP, and ServerFault) we derived the entropy of each user throughout their lifecycles, and thus each individual lifecycle period, based on the in-degree, out-degree and term distributions. We then recorded the mean of these entropy values over each lifecycle period, thereby providing an assessment of the general changes that users go through. Figure 1 presents the lifecycle period entropies for the different user properties and platforms, indicating differences between both. For instance, for users' in-degree we observe a slight increase for SAP and Facebook indicating that towards later portions of the lifecycle users tend to contact a wider array of people: this could be a symptom of users being replied to more often by the community's members as they become known. For out-degree we find a similar effect with an increase for SAP and Facebook indicating that users tend to communicate with more people over time, again this is attributable to users replying to more people's threads in order to help them with their requests.

When examining the distribution of terms, and thus the lexical dynamics of users, we find a degradation in entropy over time, indicating that users use less diverse language as they develop in the community (for SAP and Facebook), however Server Fault users remain relatively stable. This could be due to the relatively minor interaction effects that take place on ServerFault: users largely lurk on the platform to seek answers to questions, and thus do not contribute unless it is necessary (i.e. they feel that their expertise is sufficient to answer a question or that a new question is required), as a result it is likely that users have an implicit understanding of how one should formulate a post and thus the language that should be used.



(a) In-degree (b) Out-degree (c) Lexical

Figure 1. Entropies of lifetime-stage distributions formed from users' in-degrees, out-degrees and lexical terms.

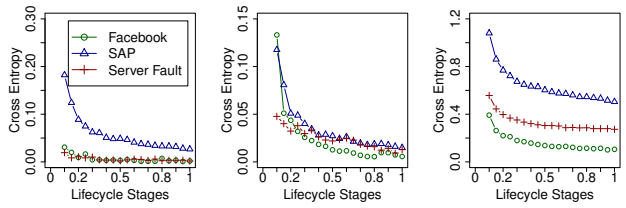
2) Historical Contrasts (Period Cross-Entropy): A limitation with focusing on time-period specific snapshots is

[3]We use $p \rightarrow q$ to denote message q replying to message p.

that we eschew time-comparative assessments of how a user is changing relative to earlier properties. To inform such cross-period assessment we examined the users' in-degree, out-degree and term distributions across lifecycle periods by computing the cross-entropy of one probability distribution with respect to another distribution from an lifecycle period, and then selecting the distribution that minimises cross-entropy. Assuming we have a probability distribution (P) formed from a given lifecycle period ($[t, t']$), and a probability distribution (Q) from an earlier lifecycle period, then we define the cross-entropy between the distributions as follows:

$$H(P, Q) = -\sum_{x} p(x) \log q(x) \tag{5}$$

In the same vein as the earlier entropy analysis, we derived the period cross-entropy for each platform's users throughout their lifecycles and then derived the mean cross-entropy for the 20 lifecycle periods. Figure 2 presents the cross-entropies derived for the different platforms and user properties. We observe that for each distribution and each platform cross-entropies reduce throughout users' lifecycles, suggesting that users do not tend to exhibit behaviour that has not been seen previously. For instance, for the in-degree distribution the cross-entropy gauges the extent to which the users who contact a given user at a given lifecycle stage differ from those who have contacted him previously, where a larger value indicates greater divergence. We find that consistently across the platforms, users are contacted by people who have contacted them before and that fewer *novel* users appear. The same is also true for the out-degree distributions: users contact fewer new people than they did before. This is symptomatic of community platforms where despite new users arriving within the platform, users form sub-communities in which they interact and communicate with the same individuals. Figure 2(c) also demonstrates that users tend to reuse language over time and thus produce a gradually decaying cross-entropy curve.



(a) In-degree (b) Out-degree (c) Lexical

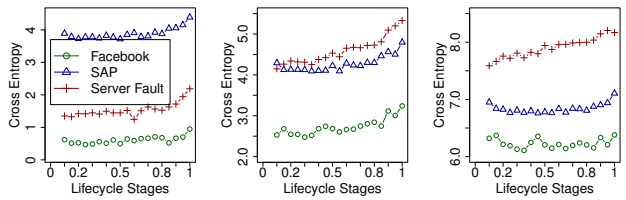
Figure 2. Cross-entropies derived from comparing users' in-degree, out-degree and lexical term distributions with previous lifecycle periods. We see a consistent reduction in the cross-entropies over time.

3) Community Contrasts (Community Cross-Entropy): For the third inspection of user lifecycles and how user properties change, we examined how users compare with the platform in which they are interacting over the same time interval. We used the in-degree, out-degree and term distributions and compared them with the same distributions derived globally over the same time periods. For the global probability distributions we used the same means as for forming user-specific distributions, but rather than using the set of posts that a given user had authored (P_{u_i}) to derive the probability distribution, we instead used all posts. For instance, for the global in-degree distribution we used the frequencies of received messages for all users. Given the discrete probability distribution of a user from a time interval ($P_{[t,t']}$), and the global probability distribution over the same time interval ($Q_{[t,t']}$), we derived the cross-entropy as above between the distributions. ($H(P_{[t,t']}, Q_{[t,t']})$).

As before we derived the community cross-entropy for each platform's users over their lifetimes and then calculated the mean community cross-entropy for the lifecycle periods. Figure 3 presents the plots of the cross-entropies for the in-degree, out-degree and term distributions over the lifecycle periods. We find that for all platforms the community cross-entropy of users' in-degree increases over time indicating that a given user tends to diverge in his properties from users of the platform. For instance, for the community cross-entropy of the in-degree distribution the divergence towards later parts of the lifecycle indicates that users who reply to a given user differ from the repliers in the entire community. This complements cross-period findings from above where we see a reduction in cross entropy, thus suggesting that users form sub-communities in which interaction is consistently performed within (i.e. reduction in new users joining). We find a similar effect for the out-degree of the users where divergence from the community is evident towards the latter stages of users' lifecycles. The term distribution demonstrates differing effects however: for Facebook and SAP we find that the community cross-entropy reduces initially before rising again towards the end of the lifecycle, while for Server Fault there is a clear increase in community cross-entropy towards the latter portions of users' lifecycles suggesting that the language used by the users actually tends to diverge from that of the community in a linear manner. This effect is consistent with the findings of Danescu et al. [2] where users adapt their language to the community to begin with, before then diverging towards the end.

V. Mining Lifecycle Trajectories

Inspecting how communities of users develop we have concentrated on assessments at the *macro-level* on each platform, examining how the social dynamics and lexical dynamics of communities of users have changed over time. We now turn to examining how *individual* users evolve throughout their lifecycle periods. Understanding how individual users develop over time in online community platforms allows for churners to be predicted, as we shall demonstrate in the following section through our experiments, and also



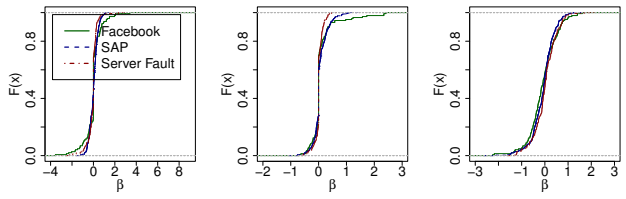
(a) In-degree (b) Out-degree (c) Lexical

Figure 3. Cross-entropies derived from comparing users' in-degree, out-degree and lexical term distributions the community platform over the same time periods. We see a increased divergence towards the end of lifecycles.

informs how online community platforms differ in terms of user development. We model the development of users throughout their lifecycles by mining the lifecycle trajectories along different user properties (in-degree, out-degree, terms) and development measures (entropy, period-cross-entropy, community-cross-entropy). Mining is performed by selecting a suitable model to represent changes in user properties based on community development, before then setting the model for individual users. We begin this section by explaining entropy trajectories and the mining process.

A. Modelling Entropy Trajectories

Our prior analysis of the entropy of user properties indicated that users exhibited a generally stable entropy throughout their lifecycle periods. Therefore we chose the linear regression model as a suitable model for the development of user properties, setting the explanatory variable to be the lifecycle period of the user and the response variable to be the user property's entropy. In modelling entropy development we can characterise each user using the slope (β) of the model, thus indicating the rate of change of entropy throughout the lifecycle periods. We induced user-specific entropy models for each platform's users and then examined the cumulative frequency distribution of the β-values for the different user properties and platforms, these are shown in Figure 4.



(a) In-degree (b) Out-degree (c) Lexical

Figure 4. Cumulative frequency distributions of linear regression models' β coefficients for the entropy of user property distributions.

The plots indicate that each platform and each user property has a distribution with a mean slope being close to 0, with greater variance in the slopes of users for their lexical entropy. This indicates that for certain users their entropy reduces over time, while for others it tends to increase. We omit the coefficient of determinant (R^2) plots here due to brevity, however they show that for >80% of users $R^2 > 0.4$.

B. Modelling Cross-Period Trajectories

Inspection of the period cross-entropy values obtained earlier, by deriving the minimum cross-entropy when comparing user properties with past properties, indicated clear decreasing trends. That is, users converge on their past behaviour over time. This suggests that an exponential decay model would be suitable for describing such reductions throughout user's lifecycles. Applying such a model requires that users reduce in their period cross-entropy values over time. To examine whether this was indeed the case we defined the measure δ_{u_i} that returns the average proportional change value in period cross-entropy for a given user throughout their lifecycles, letting $f(u_i, [t, t'])$ denote a function that returns the period cross-entropy of an arbitrary user property (e.g. in-degree) for a given user and time interval:

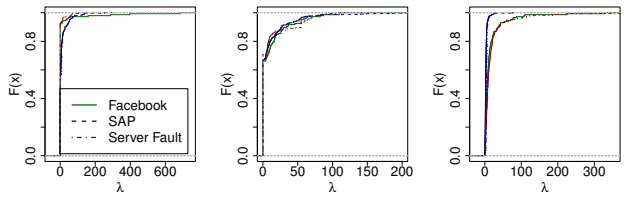
$$\delta_{u_i} = \frac{1}{|T|-1} \sum_{\substack{[t,t'],[t',t'']\in T, \\ t<t'<t''}} \frac{f(u_i,[t,t']) - f(u_i,[t',t''])}{f(u_i,[t,t'])} \tag{6}$$

By deriving the distribution of average proportional change values (δ) across the different platforms and user properties we examined the proportion of users for whom the average change was greater than 0, and thus indicating decay overall. We found that for all tested measures, all users had an average proportional change value of greater than 0, thus suggesting the suitability of a decaying growth model. The exponential decay model requires one parameter to be provided λ that defines the decay rate of a given value x (e.g. in-degree period cross-entropy) over time, where $\lambda = 1/\bar{x}$. We defined the lifecycle period for the exponential model using an integer value $s = \{1, 2, \ldots, 20\}$, hence $[t_0, t_{0.05}] \equiv s_1$, and then defined the exponential decay model as follows, letting $f(s, u_i)$ be a function that returns the period cross-entropy of an arbitrary feature (in-degree, out-degree, terms) for a given user and lifecycle period:

$$g(s, u_i) = f(s_1, u_i)e^{-\lambda s} \tag{7}$$

As we induce a per-user parameter, and thus derive a model for each user, we can characterise the decay rate of users along different properties and examine how they vary. In Figure 5 we plotted the cumulative distribution of decay rates (λ) for users of the three platforms and the three user properties. We found that every distribution had a right-skew indicating that users tend to have low decay rates, while some users have a large decay rate - indicating that these users tend to converge on their prior behaviour a lot quicker

(i.e. forming new communities quickly and converging on the same terms).



(a) In-degree (b) Out-degree (c) Lexical

Figure 5. Cumulative frequency distributions of the decay rates for users' in-degree, out-degree and lexical distributions when compared against prior time periods.

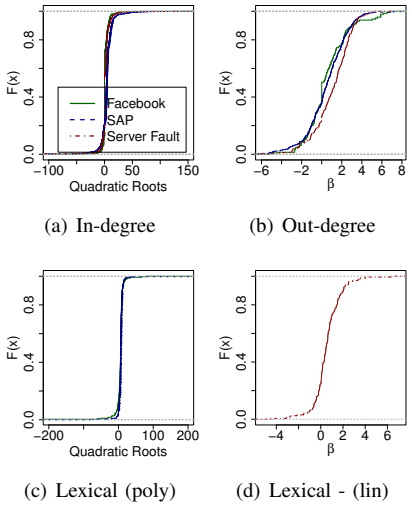
C. Modelling Cross-Community Trajectories

Inspection of the community cross-entropy values across the three user properties identified differences between the platforms and properties. We found that the in-degree distribution of users reduced after the user joined the system, before then increasing again towards the end of their life cycles, thus suggesting that a quadratic function could explain this trajectory, and that the same model could be used for the term distribution's community cross-entropy for Facebook and SAP. To derive such a model we used polynomial regression with degree $n = 2$, given that the characteristic function of development appears to be quadratic, setting the explanatory variable to be the lifecycle period of the user and the response variable to be the user property (i.e. in-degree or term distribution) community cross-entropy. Formally this is defined as follows:

$$y = a_0 + a_1 x + a_2 x^2 \tag{8}$$

By factorising the above expression we derive at most two roots (a^1 and a^2) that characterise the development of a given user based on a the selected property and community. For the out-degree distribution we found that the community cross-entropies increased in a relatively linear manner for each platform. Therefore we used a linear regression model, as in the previous section, with the lifecycle stage of the user as the explanatory variable and the community cross-entropy as the response variable. We then used the slope of the model, the β coefficient, to characterise the development of the user in terms of the distribution of users that they reply to relative to how the community is behaving as a whole. We found that $>73\%$ of users $R^2 > 0.4$ for the linear and quadratic regression models.

Figure 6(a) and Figure 6(b) presents the distribution of the quadratic roots and β coefficients respectively. Both distributions are centred around 0 and have variances in their distribution that appear to be normal. This suggests that while most users' out-degree distributions tend to diverge with the community (positive slope), certain users' distributions actually reduce over time such that they tend to mimic the communicative behaviour of the community as a whole. The same is true for the quadratic regression model (variance in the trajectory curves).



(a) In-degree (b) Out-degree

(c) Lexical (poly) (d) Lexical - (lin)

Figure 6. Cumulative frequency distributions of community cross-entropy models' features for: in-degree, the roots of quadratic regression models for all three platforms; out-degree, the linear regression β coefficient; and lexical, the roots for the quadratic regression model for Facebook and SAP and the linear regression β coefficient for Server Fault.

For the lexical distribution we found differences between the platforms in the shape of the community cross-entropy curve. For Facebook and SAP we found that the community cross entropy of users' lexical distributions reduced primarily before then increasing towards the end of the lifecycle, indicating that they use language which is similar to the community to begin with and adapt to that language throughout their lifecycle before then using language which is distinct towards the end. While for Server Fault we found users' lexical distribution diverged from the platform in a linear fashion throughout their life cycle. Therefore we decided to use two different models to capture these different functions: quadratic regression for Facebook and SAP, and linear regression for Server Fault. Figure 6(c) and 6(d) show the distribution of the quadratic model's roots and the linear regression model's slope coefficient respectively. As with the previous models, the mean of both the roots and slope coefficients are centred around 0, and each distribution has normally distributed variance indicating that users differ in their development relative to the community.

VI. Lifecycle Application: Predicting Churners

Understanding the lifecycle trajectories of users provides a means to examine and categorise users based on their exhibited behaviour. One of the central motivations behind our work was to enable the identification, and hence the

subsequent interpretation, of *churners* in online communities. Churners present a serious issue for community managers and hosts as the leaving of certain users can have a detrimental effect on the community (i.e. experts leaving a question-answering community can cause an increase in unanswered queries).

In this section we define churn prediction as a binary classification task and use the previously examined indicators of lifecycle trajectories to predict whether a user is a churner or not. As we confine user lifecycle periods from the start of their lifecycle to the end we use the trajectories mined from this period to characterise how users develop. We define churners as any user who posts for the last time before the final 10% of the time window of our datasets, cutoff points are: 2012-07-09 for Facebook, 2010-05-11 for SAP, and 2010-12-23 for ServerFault. Our dataset is of the following form: $D = \{(\mathbf{x}_i, y_i)\}$, where y_i denotes the class label of the user from one of two values: $y \in \{0, 1\}$,[4] while $\mathbf{x}_i$ denotes an 11-element $\mathbb{R}$-valued feature vector for either a Facebook or SAP user, and a 10-element feature vector for a Server Fault user - given that we use a linear regression model for each user's lexical community cross-entropy development. We model the feature vector of each user using the trajectory indicators from the previous section, in short Table II defines our set of features where we place each within a set depending on the dynamics it captures.

Table II
FEATURES USED FOR THE CHURN PREDICTION EXPERIMENTS. THE INDICATORS OF LIFECYCLE TRAJECTORIES ARE USED TO CHARACTERISE USER EVOLUTION ALONG THE DIFFERENT USER PROPERTIES.

Property	Indicator	Model Feature(s)	Platform
In-degree	Period Entropy	Linear Regression β	All
	Period Cross-Ent	Exponential Decay λ	All
	Comm' Cross-Ent	Quad' Regress' a^1, a^2	All
Out-degree	Period Entropy	Linear Regression β	All
	Period Cross-Ent	Exponential Decay λ	All
	Comm' Cross-Ent	Linear Regression β	All
Lexical	Period Entropy	Linear Regression β	All
	Period Cross-Ent	Exponential Decay λ	All
	Comm' Cross-Ent	Quad' Regress' a^1, a^2	Fb, SAP
	Comm' Cross-Ent	Linear Regression β	SF

A. Prediction Model Definition

The observed feature vector of user u_i ($\mathbf{x}_i$) contains the indicator trajectories of the user along the different properties. We use the logistic regression model to predict the conditional probability of user u_i churning as follows:

$$Pr(Y = 1 \mid \mathbf{x}_i) = \frac{1}{1 + e^{-\boldsymbol{\beta}^\intercal \mathbf{x}_i}} \tag{9}$$

The model's coefficients ($\boldsymbol{\beta}$) define the weight attached to each identity trajectory feature within the linear model ($f(i) = \boldsymbol{\beta}^\intercal \mathbf{x}_i$). In order to derive the model's coefficients we use maximum likelihood fitting through the R statistical software package[5] to select the maximum likelihood estimation $\hat{\boldsymbol{\beta}}$ of the model's coefficients. Following fitting, the derived model is used to predict the churn probability of each user within the test dataset.

B. Experimental Setup

For our experiments we first standardised the datasets by combining the test (20%) and training (80%) datasets together and setting each indicator feature to have 0 mean and a standard deviation of 1, we then divided the dataset again into the respective test and training splits maintaining the same instances as before. We wanted to test the effects of observing different user properties and development dynamics on churn prediction. We therefore tested each user property in isolation, for instance using the in-degree property and the entropy, period cross-entropy and community cross-entropy trajectory indicators; and then each development model in isolation, for instance using the entropy model and examining in-degree, out-degree, and term distributions; finally we combined all features together within a single model. In doing so we could isolate any effects of key features on prediction performance, and thus inform model selection for specific platforms (i.e. identifying the best performing model for Facebook, SAP and Server Fault).

As we used the logistic regression model for our prediction model we are provided with a function whose co-domain is a churn probability value for a given user within the closed interval $[0, 1]$. Therefore we evaluated the performance of each induced model using two evaluation measures: (i) precision@k ($\bar{P}$), and (ii) area under the receiver operator characteristic curve (AUC). To derive precision@k we ranked the users by their churn probability according to the induced model and then assessed the precision of the top-k ranks, setting $k = \{1, 5, 10, 20, 50, 100\}$, and taking the mean of these precision values. This assesses the extent to which the upper portion of the predicted churners are correct. We used the baseline measure of the probability of a randomly selected user being a churner, thus corresponding to the probability of success in a single Bernouilli trial (setting $p = |churners|/|D_{test}|$). To derive the area under the receiver operator characteristic curve we varied the confidence of an indicator function ($f(\mathbf{x})$) through discrete settings of confidence bounds $\sigma = \{0, 0.05, \ldots, 0.95, 0.1\}$, thereby setting the class label for given instance ($\mathbf{x}$) as follows:

$$f(\mathbf{x}) = \begin{cases} 1, & \text{if } Pr(Y = 1 \mid \mathbf{x}_i) > \sigma \quad (10a) \\ 0, & \text{otherwise} \quad (10b) \end{cases}$$

For each different setting of σ we measured the true positive rate (TPR/recall) and the false positive rate (FPR), and from these measures plotted the receiver operator characteristic (ROC) curve. A model which maximises the area

[4] 1 indicating churner, 0 not.

[5] http://www.r-project.org/

under this curve (AUC) is preferable (thus achieving a value of 1), where the baseline for this measure is 0.5.

C. Results

Table III presents the performance of the different models over the three platforms, showing variation in the optimum model and evaluation measures. Interestingly, we find that the use of all features combined together does not yield the best model for any of the tested platforms. For Facebook the results indicate that the prediction model using community cross entropy indicators performed best in terms of both $\bar{P}$ and AUC. We tested the difference between this model and next best performing model (Full) using a Mann-Whitney test and found the difference to be significant (at the 5% level).

For SAP we found differences in the best performing model according to the evaluation measure used: in-degree for AUC and lexical features for $\bar{P}$. These differences indicate that concentrating on top ranks and thus informing detection of churners with high-levels of confidence can be best achieved by assessing the term distributions of users and their lexical dynamics, while for preferring recall the use of in-degree distributions is preferable.

Turning now to Server Fault, the results also indicate differences in the best model depending on the evaluation measure: period cross-entropy achieves the best AUC score, while the lexical features yield the highest $\bar{P}$. This latter result indicates similarities between SAP and Server Fault, both of which are Question-Answering community platforms, in that that the terms used by users are salient indicators of churners and for guaranteeing high precision when identifying who will churn.

D. Churn Patterns

Up to this point we have concentrated on predicting churners from online communities, we now turn to inspecting the qualities that discern churners from other users. One of the advantages of using a logistic regression model as our churn prediction model is its provision of coefficients that can be inspected to understand the impact of individual features on the likelihood of a user churner, that is: by inspecting the coefficient of a given feature within the model we can understand how a change in that feature (increase/decrease) would affect the churn probability.

Table IV shows the features and coefficients from the best performing model, based on maximising the AUC, from the previous section. For both Facebook and SAP we find that the roots of the quadratic regression model for the community cross-entropy of the in-degree distributions should be reduced, suggesting that the curve has a vertex which appears prior to the time interval analysed and that the trajectory of a churner has a steep gradient to towards the end of their lifecycle. This indicates that churners exhibit divergent social networks in comparison to the platform.

Table III
PRECISION@K ($\bar{P}$) AND AREA UNDER THE RECEIVER OPERATOR CHARACTERISTIC CURVE (AUC) VALUES FOR FACEBOOK, SAP AND SERVER FAULT WHEN TESTING DIFFERENT: (I) USER PROPERTIES, (II) DEVELOPMENT INDICATORS, (III) ALL FEATURES TOGETHER.

Platform	Feature	$\bar{P}$	AUC
Facebook	Entropy	0.761	0.500
	Period Cross Entropy	0.624	0.485
	Community Cross Entropy	**0.791**	**0.617**
	In-degree	0.648	0.511
	Out-degree	0.781	0.570
	Lexical	0.681	0.557
	Full	0.730	0.573
	Baseline	0.629	0.500
SAP	Entropy	0.434	0.549
	Period Cross Entropy	0.321	0.568
	Community Cross Entropy	0.334	0.549
	In-degree	0.351	**0.592**
	Out-degree	0.250	0.503
	Lexical	**0.438**	0.539
	Full	0.363	0.539
	Baseline	0.342	0.500
Server Fault	Entropy	0.392	0.526
	Period Cross Entropy	0.300	**0.555**
	Community Cross Entropy	0.352	0.538
	In-degree	0.232	0.475
	Out-degree	0.293	0.512
	Lexical	**0.459**	0.546
	Full	0.421	0.554
	Baseline	0.319	0.500

We also find differences between SAP and Server Fault in terms of the period cross-entropy trajectory for the in-degree distributions: as we induced exponential decay models for each of these platforms, the decay rate indicates that for SAP this rate should be increased for churners, suggesting their in-degree distributions, and the extent to which they are contacted during one time period relative to their past communications, reduces at a much faster rate than on ServerFault.

Table IV
BEST PERFORMING PREDICTION MODEL COEFFICIENTS FOR FACEBOOK (COMMUNITY CROSS-ENTROPY), SAP (IN-DEGREE) AND SERVER FAULT (PERIOD CROSS-ENTROPY). ALL FEATURES ARE SIGNIFICANT WITHIN THEIR RESPECTIVE MODELS ($\alpha < 0.05$)

Feature	Facebook	SAP	Server Fault
In-degree Entropy β	-	0.0532	-
In-degree Period Cross-Ent λ	-	0.0139	-0.1826
In-degree Comm' Cross-Ent a^1	-0.1057	-0.1878	-
In-degree Comm' Cross-Ent a^2	-0.0510	-1.5104	-
Out-degree Comm' Cross-Ent β	0.3173		
Out-degree Period Cross-Ent λ	-	-	0.0210
Lexical Period Cross-Ent λ	-	-	0.0557
Lexical Comm' Cross-Ent a^1	0.3253		-
Lexical Comm' Cross-Ent a^2	-0.0541		-

VII. DISCUSSION AND FUTURE WORK

Prior work on social network evolution by Panzarasa et al. [6] and Miritello et al. [1] found that users' social networks tend to a limit in terms of their *communication capacity*. Our examination of the development of users in terms of their in-degree and out-degree distributions relative to earlier lifecycle periods (period cross-entropy) demonstrated

similar findings, where users converge on communicating with a fixed number of users. Comparison against the social network of the platform over the same time periods showed that users do not tend not to add any new connections, thus settling on their own distinct network. This supports the early identity development work of Erikson [3] which theorised that people are susceptible to alterations within the adolescent stages of their lives, taking on influence from their peers and thus adjusting their interests/tastes to reflect the individuals around them. We see effects to support this theory in the community cross-entropy plots where in the early-to-mid portions of users' lifecycles there is a tendency for the user to become more similar to how the community is interacting as a whole (decrease for in-degree and lexical distributions). Recent work by Hong et al. [11] has demonstrated the potential for stage-based recommendations (considering how a customer's tastes develop over time), therefore future work will examine the use of lifecycle periods for building user networks for collaborative filtering and the effect of this on recommendation performance.

Our churn prediction experiments have explored the applicability of different user properties and lifecycle trajectories. Recent work by Danescu et al. [2] examined the predictive performance of lexical features on churn prediction, finding that the incorporation of the term cross-entropy (derived by comparing a user's term distribution with that of the community from a window of 12-months either side of a given lifecycle period) improved prediction performance. Our findings suggest that the incorporation of lexical dynamics achieves good performance in terms of precision for SAP and ServerFault, while the incorporation of social dynamics achieves improved performance in terms of recall-focussed AUC. This suggests that the prediction model's applicability is context-dependent: in certain cases lexical information is preferable, while in others the communications associated with a user are preferred.

One limitation with our approach is the reliance on knowledge *a priori* of churn lifecycle stages: our future work will seek to apply the approach in a setting where such stages are not known, for instance given arbitrary date points. This would also allow for more advanced churn prediction tasks to be performed including predicting in how many days a given user will post for the last time. Such tasks would also allow for comparative evaluation against other methods such as survival analysis.

VIII. Conclusion

In this paper we have presented a means to model the development of users within online community platforms based on both social and lexical dynamics. We have complemented existing work [1] and Danescu et al. [2] by assessing the development of users: (i) over individual lifecycle periods, (ii) relative to their past states, and (iii) relative to the community in which they are interacting. We first examined the overall development of users before then selecting suitable development models and induced models specific to individual users. In doing so we could then mine *lifecycle trajectories* describing the evolution of users along different properties and development indicators.

The utility of lifecycle trajectories, and thus the characterisation of user development, is such that we can accurately discern churners from non-churners across the three online community platforms that we studied. This indicates that by considering how users develop relative to their past behaviour and the community in which they are interacting we can identify who will churn more accurately than by looking at isolated snapshots of user development.

References

[1] G. Miritello, R. Lara, M. Cebrián, and E. Moro, "Limited communication capacity unveils strategies for human interaction," Apr. 2013. [Online]. Available: http://arxiv.org/abs/1304.1979

[2] C. Danescu-Niculescu-Mizil, R. West, D. Jurafsky, J. Leskovec, and C. Potts, "No country for old members: User lifecycle and linguistic change in online communities," in *Proceedings of the World Wide Web Conference*, 2013.

[3] E. Erikson, "Ego development and historical change," *The Psychoanalytic Study of the Child*, vol. 2, pp. 359–396, 1959.

[4] J. McAuley and J. Leskovec, "From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews," in *Proceedings of World Wide Web Conference*, 2013.

[5] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 462–470.

[6] P. Panzarasa, T. Opsahl, and K. M. Carley, "Patterns and dynamics of users' behavior and interaction: Network analysis of an online community," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 5, pp. 911–932, May 2009. [Online]. Available: http://dx.doi.org/10.1002/asi.v60:5

[7] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 44–54.

[8] S. R. Kairam, D. J. Wang, and J. Leskovec, "The life and death of online groups: predicting group growth and longevity," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 673–682.

[9] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song, "Evolution of social-attribute networks: Measurements, modeling, and implications using google+," *CoRR*, vol. abs/1209.0835, 2012.

[10] K. S. K. Chung, M. Piraveenan, and S. Uddin, "Community evolution and engagement through assortative mixing in online social networks," *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, vol. 0, pp. 724–725, 2012.

[11] W. Hong, L. Li, and T. Li, "Product recommendation with temporal dynamics," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12 398–12 406, Nov. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2012.04.082