

# ABSoNeS: Agent Based Social Network Simulation

Simone Ciccolella<sup>1</sup> and Daniele Bellani<sup>2</sup>

<sup>1</sup>*email*

<sup>2</sup>*email*

13 maggio 2017

## TO DO

- Questione **omofilia**
- Definire costati di proporzione
- Definire alta e bassa probabilità e attività

## 1 Modellazione

Per la creazione del modello deriviamo delle costanti che saranno sempre vere:

**Fasi Temporalì (FT)** Dividiamo la giornata in 12 fasi che rappresentano 2 ore l'una in cui ogni utente ha un personale valore di **activity** che rappresenta il suo utilizzo di Twitter. Tali fasi sono così suddivise:

- 4 fasi con **activity** pari a 0 corrispondente a 8 ore di sonno, ovvero la quantità consigliata (e quasi mai rispettata).
- 4 fasi con **activity** bassa, corrispondente a 8 ore di lavoro
- 4 fasi con **activity** elevata che rappresenta 8 ore di tempo libero in cui l'utente ha una attività maggiore su Twitter

Al momento ignoramo il weekend, ma cercheremo di definirlo e implementarlo nelle prossime versioni. [•**SC**: definire BASSA e ELEVATA attività •]

**Topic** ovvero i possibili argomenti di interesse esistenti nel modello perciò abbiamo una lista **topic** =  $(t_1, \dots, t_n)$  definiti a priori e immutabili. Non vengono mai utilizzati direttamente, ma vengono utilizzati come indici di riferimento dunque non è necessario implementarli realmente.

## 1.1 Utenti

Gli utenti sono rappresentati dai nodi della rete. Vediamo ora dunque in che modo viene generato ogni nodo:

**Personal Interest (PI)** è una lista  $PI = (p_1, \dots, p_n)$  dove  $p_i \in [0, 1]$  [●SC: valutare inclusione degli estremi ●] indica la probabilità di interesse del nodo rispetto al **topic**  $i$  generata casualmente. Al momento tale valore è immutabile. [●SC: modellare valore variabile ●]

**Timezone (TZ)** ovvero una lista  $TZ = (tz_1, \dots, tz_{12})$  dove i  $tz_i$  sono generati in accordo con le **FT** generati come segue:

- Genere  $1 \leq i \leq 12$  casualmente
- Imposto le ore di sonno  $tz_j = 0$  per  $j = i, \dots, i + 3$
- Imposto le ore di lavoro  $tz_j = \text{bassa activity}$  per  $j = i + 4, \dots, i + 7$  [●SC: definire ●]
- Imposto le ore di tempo libero  $tz_j = \text{alta activity}$  per  $j = i + 8, \dots, i + 11$  [●SC: definire ●]

Tutte le precedenti operazioni di indici sono da considerarsi *mod* 12. Il motivo di questa scelta è quello di simulare sia diversi orari per le persone, sia diversi fusi orari.

Per poter fare operazioni sugli utenti abbiamo inoltre bisogno delle seguenti definizioni:

**Followers** per ogni utente  $U$  definiamo come **followers**( $U$ ) l'insieme dei followers di  $U$ .

**Following** similmente **following**( $U$ ) indica l'insieme degli utenti seguiti da  $U$ .

**Direct Tag (dtag)** **dtag**( $U, T$ ) è l'insieme dei tweets che contengo un tag all'utente  $U$  generati al tempo  $T$ . Sarà spiegato meglio nella sezione riguardante i tweet [1.3].

**Interessi** per ogni utente  $U$  **interest**( $U$ ) è l'insieme dei **topic** che interessano a  $U$ .

**Field Of View (FOV)** **FOV**( $U, T$ ) rappresenta l'insieme delle notizie visualizzate dall'utente  $U$  al tempo  $T$ , con  $FOV \subseteq \text{tweet}(\text{following}(U), T_i) \cup \text{retweet}(\text{following}(U), T_j) \cup \text{dtag}(U, T_l)$  per  $T_{i,j,l} \in \bar{T}$  lista di tempi non maggiori al tempo attuale  $T$ . Questo insieme è definito sia per una questione computazionale, ma anche per un motivo reale in quanto è difficile che un utente nell'arco della giornata riesca a vedere tutti i tweets e i retweets degli utenti che segue e i tweet che in cui risulta direttamente *taggato*. Questa affermazione diventa sempre più ragionevole al crescere della popolarità di un utente.

Le relazioni tra gli utenti sono descritti dagli archi tra di essi.

**Edges** un arco tra due utenti  $U_1$  e  $U_2$ , scritto come  $(U_1, U_2)$  rappresenta la relazione di following tra il primo ed il secondo. Di conseguenza, ovviamente, risulta che  $U_1 \in \text{followers}(U_2)$  e simmetricamente  $U_2 \in \text{followers}(U_1)$

**Attachment** ad un arco  $e_j \in U \times U$  è associato un valore  $\text{attachment}(U_1, U_2) \in [0, 1]$  che rappresenta l' *attaccamento* di  $U_1$  a  $U_2$ , più questo valore si avvicina a 0 più la probabilità che  $U_1$  smetta di seguire  $U_2$  aumenta e viceversa. Quando il nodo viene creato il valore di **attachment** è relativamente alto ( $\sim 0.8$ ) in quanto ci si aspetta che un utente non smetta di seguire un altro utente poco dopo aver iniziato a seguirlo.

## 1.2 Probabilità

In tutto il modello sono definite globalmente le probabilità di compiere una determinata azione.

**Tweet** ogni utente  $U$  ad ogni tempo  $T$  ha la possibilità di creare un tweet secondo la seguente probabilità:

$$(1) \quad P_{\text{tweet}} = \alpha T Z(T) \frac{|\text{followers}(U)|}{|Users|}$$

Questo perchè ci si aspetta che un utente popolare sia più attivo di uno sconosciuto, per mantenere il suo livello di popolarità.

**Retweet** similmente al precedente un utente  $U$  al tempo  $T$  ha una possibilità di effettuare un retweet come:

$$(2) \quad P_{\text{retweet}} = \beta T Z(T) \frac{|\text{followers}(U)|}{|Users|}$$

Risulta da valutare la possibilità che un utente possa fare più retweet rispetto al numero dei tweet, in quanto essa è un'operazione decisamente meno impegnativa e perciò, intuitivamente, con una più elevata probabilità di accadere.

**Tag** ogni tweet ha una possibilità di contenere un tag diretto ad un altro utente  $V$  con probabilità

$$(3) \quad P_{\text{tag}} = \gamma \frac{\text{follower}(U) \text{followers}(V)}{|Users|^2}$$

Ovvero ci si aspetta che il **dtag** sia proporzionale alla popolarità di entrambi gli utenti coinvolti.

### 1.3 Attività sociali

Per prima cosa definiamo gli oggetti che riguardano le attività sociali:

**Tweet** Il tweet dell'utente  $U$  al tempo  $T$  è definito come  $\text{tweet}(U, T) = (j, \text{likability}, \text{dislikability}, \text{dtag}, U)$  dove:

- $j$  è il **topic** su cui il tweet verte
- $\text{likability} \in [0, 1]$  indica la probabilità di quanto il tweet possa piacere agli utenti a cui interessa il **topic**  $j$ .
- $\text{dislikability}$  rappresenta la probabilità di quanto il tweet possa non piacere a chi non è interessato all'argomento
- $\text{dtag}$  indica l'utente  $V$  taggato nel tweet. Tale valore può anche essere nullo.
- $U$  è l'utente che ha effettuato il tweet. Necessario per l'implementazione.

**Retweet** Il retweet effettuato dall'utente  $U$  al tempo  $T$  del tweet di  $V$  al tempo  $\bar{T}$ , definito come:  $\text{retweet}(U, T) = \text{tweet}(V, \bar{T})$

**Dtag**

In base agli oggetti definiti in precedenza possiamo definire le azioni di:

**Post** Nel momento in cui l'utente  $U$  è abilitato alla creazione di un tweet allora viene generato casualmente un **topic**  $j$  su cui verterà il tweet in modo tale che un topic di interesse per  $U$  sia selezionato con maggiore probabilità. Una volta scelto il topic  $j$ :

- se  $j$  è di interesse per  $U$  ( $U.PI(j) \geq 0.5$ ) allora il **tweet** risultante avrà una **likability** elevata mentre la **dislikability** sarà casuale.
- viceversa il **tweet** generato avrà una **dislikability** alta ed una **likability** casuale.

Una volta definito il tweet viene generata la probabilità di avere un **dtag** ad un altro utente  $V$  in base alla eq. 3. Se tale tag viene generato allora il **tweet** avrà un tag all'utente  $V$ , altrimenti il post non avrà alcun **dtag**.

**Repost** Per ogni utente  $U$  viene scelta casualmente una lista di  $k$  **tweet**  $\bar{w} \in \text{FOV}(U)^k$ , successivamente per ogni  $\bar{w}_i \in \bar{w}$  viene valutata la possibilità di retweet di  $\bar{w}_i$  secondo l'equazione 2, in caso favorevole viene prodotto  $\text{retweet}(U, T)$  che sarà una lista di retweet effettuati da  $U$  al tempo  $T$ .

**Unfollow** Un utente  $U$  può decidere di smettere di seguire un utente  $V$ , grazie alla variabile **attachment** che li lega, e all'ultimo **tweet**  $W$  di  $V$  con probabilità

$$P_{\text{unfollow}}(U, V) = P(W[\text{dislikability}] \mid \text{attachment}(U, V))$$

Nel caso in cui U decida di continuare a seguire V allora  $\text{attachment}(U, V) = P_{\text{unfollow}}(U, V)$  aggiornando dunque la probabilità di unfollow a quella appena calcolata. Risulta dunque che la probabilità di unfollow dipende solo dallo stato precedente di **attachment**.  
[●SC: Markov Chain? ●]

**Continue Follow** Un utente U può avere il desiderio di continuare a seguire un altro utente V, se V ha pubblicato un tweet riguardante un topic di suo gradimento. Tuttavia tale post ha una sua **likability** che potrebbe pregiudicare, sia in positivo che in negativo, l' $\text{attachment}(U, V)$  che perciò a seguito del nuovo **tweet** W di V risulta che  $\text{attachment}(U, V) = P(W[\text{likability}] \mid \text{attachment}(U, V))$

**Nota:** Per le due azioni precedenti di **Unfollow** e **Continue Follow** bisogna considerare il caso in cui la valutazione venga effettuata rispetto ad un retweet e non ad un tweet dell'utente seguito V. [●SC: Che si fa? Si ignorano i retweet o li si considera come se fossero tweet di V, in quanto se retweetta condivide la loro opinione? ●]

**Follow** Esistono diversi tipi di possibili modalità di following:

**By Retweet (BR)** By retweet Ancora da definire. Probabilità simmetrica a **Unfollow**. [●SC: definire ●]

**Outside Factor (OF)** avviene quando un utente U comincia a seguire un nodo V per fattori esterni al Social Network, quali ad esempio nuova amicizia nella vita reale, nuovo follow su altri mezzi di comunicazioni online, ecc. [●SC: prob da definire ●]

**Active Search (AS)** un utente U ricerca un qualunque altro utente V tramite la rete sociale anche se non ha collegamenti con esso. Tale ricerca ha senso secondo il modello di omofilia [●SC: definire ●].

## 1.4 Step Simulazione

Usando il modello definito in precedenza possiamo ora descrivere gli step che avvengono nella simulazione al tempo T.

**Tweet Step (TS)** Per ogni utente U viene generato una probabilità casuale  $P_t(U, T)$  che rappresenta la sua inclinazione di produrre un **tweet** al tempo T. Se  $P_t(U, T)$  risulta conforme [●SC: conforme mi fa schifo. Qual è il termine giusto? ●] alla probabilità  $P_{\text{tweet}}$  [eq. 1] allora viene generato il **tweet**(U, T) come descritto in **Post**.

**Retweet Step (RS)** Similmente alla fase precedente, per ogni utente U viene generato una probabilità casuale  $P_r(U, T)$  che rappresenta la sua inclinazione di produrre un **retweet** al tempo T. Se  $P_r(U, T)$  risulta conforme [●SC: conforme mi fa schifo. Qual è il termine giusto? ●] alla probabilità  $P_{\text{retweet}}$  [eq. 2] allora viene generato il **retweet**(U, T) come descritto in **Repost**.

**Evaluation Step (ES)** Dopo che tutti i **tweet** e i **retweet** sono stati creati al tempo  $T$ , per ogni utente  $U$  viene generato il  $FOV(U, T)$  e per ogni  $f_i \in FOV(U, T)$  viene valutato l' **attachment**( $U, V$ ), dove  $V$  è l'autore di  $f_i$ . In base alle probabilità e azioni precedentemente descritte, l'utente  $U$  decide se smettere di seguire  $V$  (secondo **Unfollow**) o di continuare a seguire a seguire, modificando l'**attachment** come descritto in **Continue Follow**. [●**SC**: modificare in base alla considerazione dei retweet●] Nel caso in cui  $U$  stia valutando un retweet di un utente  $W$  che non segue, allora valuta la probabilità di seguirlo in base al **Following BR**. Infine viene valutata la possibilità di creare un nuovo follow per **OF** e **AS**.

Le azioni descritte in precedenza devono necessariamente essere eseguite sequenzialmente una dopo l'altra, tuttavia i singoli step possono essere facilmente parallelizzati, senza problemi di concorrenza, in quanto il post di un tweet è indipendente per ogni utente; lo stesso vale per il retweet e infine l'aggiornamento dell' **attachment** è dipendente dai tweet e i retweet effettuati **TS** e **RS**, ma i singoli aggiornamenti sono indipendenti tra gli utenti. Per questo motivo l'idea è quella di fare in modo che le azioni contenute in un step di simulazione vengano effettuate con più thread, ma comunque rispettando l'ordine descritto.