Analisi del grafo di Stack Overflow

Metodi e applicazioni per social network

Riccardo Macoratti (844553)

Indice

Indice

Stack Overflow (1)

Stack Overflow è la prima piattaforma della rete di Stack Exchange, dove è possibile porre domande riguardo argomenti di programmazione e ricevere risposte da altri utenti iscritti alla community, corredate da utili commenti di precisazione. La community è aperta in lettura, ma chiusa all'interazione.

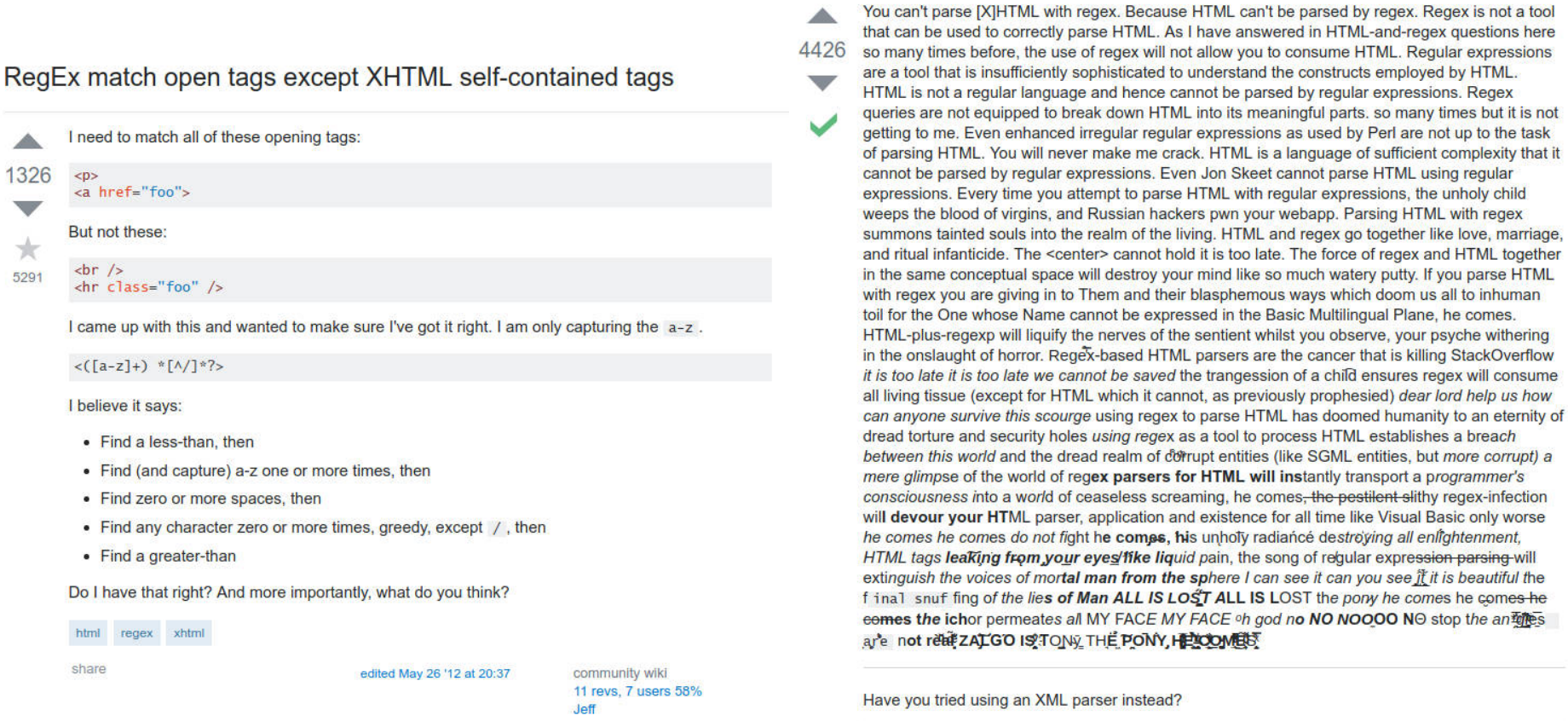
Stack Overflow (2)

La community è attiva da 8 anni e 11 mesi e conta[1]:

- **7.3 milioni** di utenti
- **14 milioni** di domande
- **22 milioni** di risposte
- **58 milioni** di commenti
- **49 mila** tag

[1] https://stackexchange.com/sites?view=list#traffic

Stack Overflow (3)



RegEx match open tags except XHTML self-contained tags

1326

5291

I need to match all of these opening tags:

```
<p>
<a href="foo">
```

But not these:

```
<br />
<hr class="foo" />
```

I came up with this and wanted to make sure I've got it right. I am only capturing the `a-z` .

```
<([a-z]+) *[^/]*?>
```

I believe it says:

- Find a less-than, then
- Find (and capture) a-z one or more times, then
- Find zero or more spaces, then
- Find any character zero or more times, greedy, except `/` , then
- Find a greater-than

Do I have that right? And more importantly, what do you think?

html regex xhtml

share

edited May 26 '12 at 20:37

community wiki
11 revs, 7 users 58%
Jeff

4426

You can't parse [X]HTML with regex. Because HTML can't be parsed by regex. Regex is not a tool that can be used to correctly parse HTML. As I have answered in HTML-and-regex questions here so many times before, the use of regex will not allow you to consume HTML. Regular expressions are a tool that is insufficiently sophisticated to understand the constructs employed by HTML. HTML is not a regular language and hence cannot be parsed by regular expressions. Regex queries are not equipped to break down HTML into its meaningful parts. so many times but it is not getting to me. Even enhanced irregular regular expressions as used by Perl are not up to the task of parsing HTML. You will never make me crack. HTML is a language of sufficient complexity that it cannot be parsed by regular expressions. Even Jon Skeet cannot parse HTML using regular expressions. Every time you attempt to parse HTML with regular expressions, the unholy child weeps the blood of virgins, and Russian hackers pwn your webapp. Parsing HTML with regex summons tainted souls into the realm of the living. HTML and regex go together like love, marriage, and ritual infanticide. The cannot hold it is too late. The force of regex and HTML together in the same conceptual space will destroy your mind like so much watery putty. If you parse HTML with regex you are giving in to Them and their blasphemous ways which doom us all to inhuman toil for the One whose Name cannot be expressed in the Basic Multilingual Plane, he comes. HTML-plus-regexp will liquify the nerves of the sentient whilst you observe, your psyche withering in the onslaught of horror. Regex-based HTML parsers are the cancer that is killing StackOverflow *it is too late it is too late we cannot be saved* the trangession of a child ensures regex will consume all living tissue (except for HTML which it cannot, as previously prophesied) *dear lord help us how can anyone survive this scourge* using regex to parse HTML has doomed humanity to an eternity of dread torture and security holes *using regex* as a tool to process HTML establishes a breach *between this world* and the dread realm of corrupt entities (like SGML entities, but *more corrupt) a mere glimpse* of the world of reg**ex parsers for HTML will ins**tantly transport a *programmer's consciousness i*nto a world of ceaseless screaming, he comes~~, the pestilent slithy~~ regex-infection will **devour your HT**ML parser, application and existence for all time like Visual Basic only worse *he comes he comes do not fight* he com~~es~~**es, hi**s unholy radiańcé de*stroying all enlightenment, HTML tags leaking from your eyes/like liquid* pain, the song of regular expre~~ssion parsing~~ will ext*inguish the voices of mortal man from the sphere I can see it can you see it it is beautiful* the `final snuf` fing *of the lies of Man ALL IS LOST* **ALL IS** LOST the *pony he comes* he com~~es he comes the~~ **ich**or permeates *all* MY FACE *MY FACE ᵒh god no* **NO NOOOOO** NΘ stop *the an*~~gles~~ are **not real ZALGO IS TONY THE PONY, HE COMES**

Have you tried using an XML parser instead?

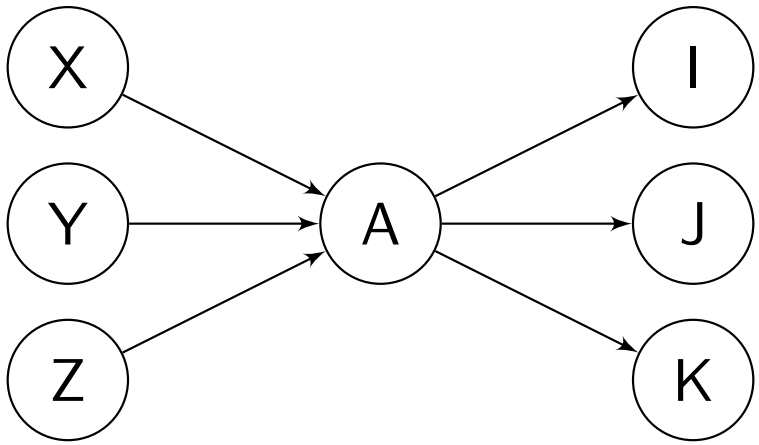
Una domanda (a sinistra), una risposta (a destra).

Indice

Struttura del grafo

Il grafo che rappresenta la struttura di Stack Overflow utilizzata per l'analisi è un grafo orientato in cui:

- i nodi rappresentano gli utenti di Stack Overflow
- gli archi rispondono alla domanda "ha risposto a"
- ogni arco ha un peso che corrisponde al numero delle volte che la sorgente ha risposto alla destinazione



Il grado in entrata è una misura delle domande, quello in uscita delle risposte.

Algoritmo di acquisizione dei dati

Come frontiera è stato utilizzato l'utente con la reputazione più alta, Jon Skeet (`id` 22656).

```
SO_Users := [22656, ...]
G := DiGraph()

for each user in SO_Users do
  questions := user.questions()
  for each question in questions do
    answers := question.answers()
    for each answer in answers do
      if answer.owner not in G do
        G.make_edge(answer.owner, user)
        SO_Users.add(answer.owner)
      end
    end
  end
end
```

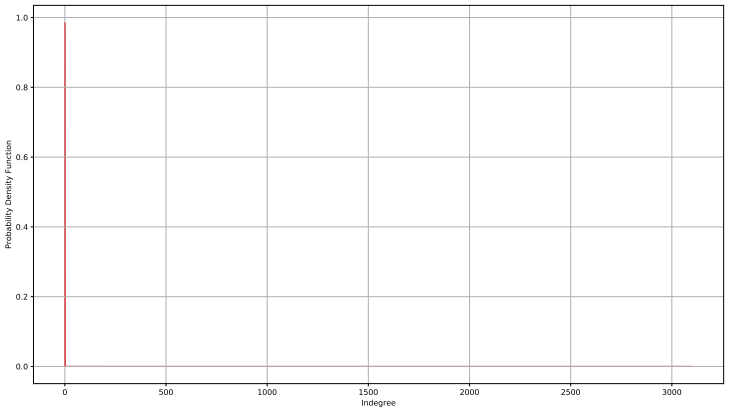
Indice

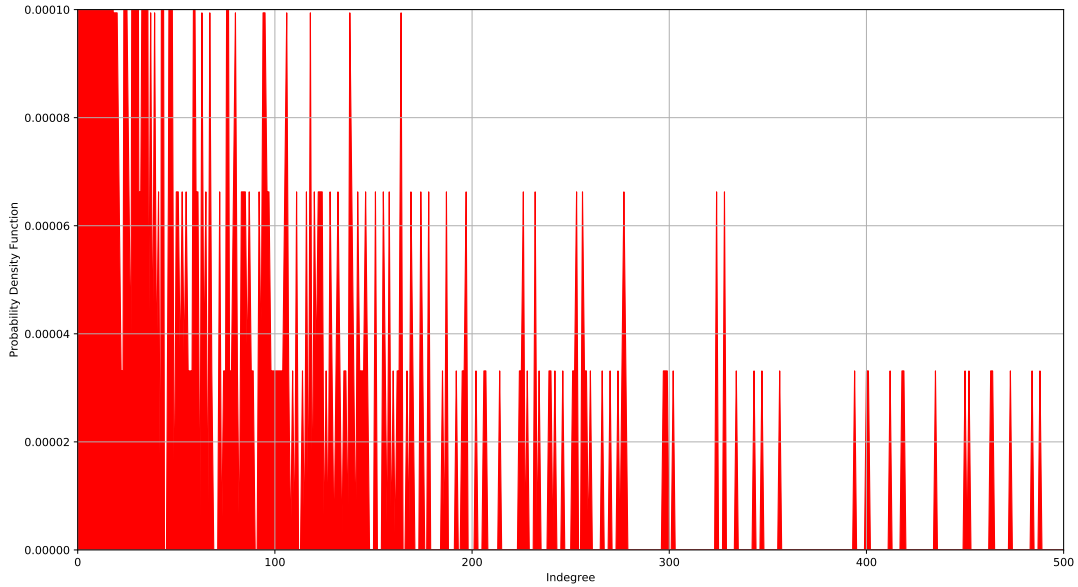
Parametri cumulativi

La rete ha **30182** nodi e **56344** archi e il grafo associato non è fortemente connesso.
La densità della rete è 0.0000618538[2].
Il grado in entrata minimo è **0** mentre quello massimo è **3102**, ed appartiene a acidzombie24. Il grado in uscita minimo è **1** mentre quello massimo è **142**, ed appartiene a Jon Skeet.
Il grado medio è **1.867**.

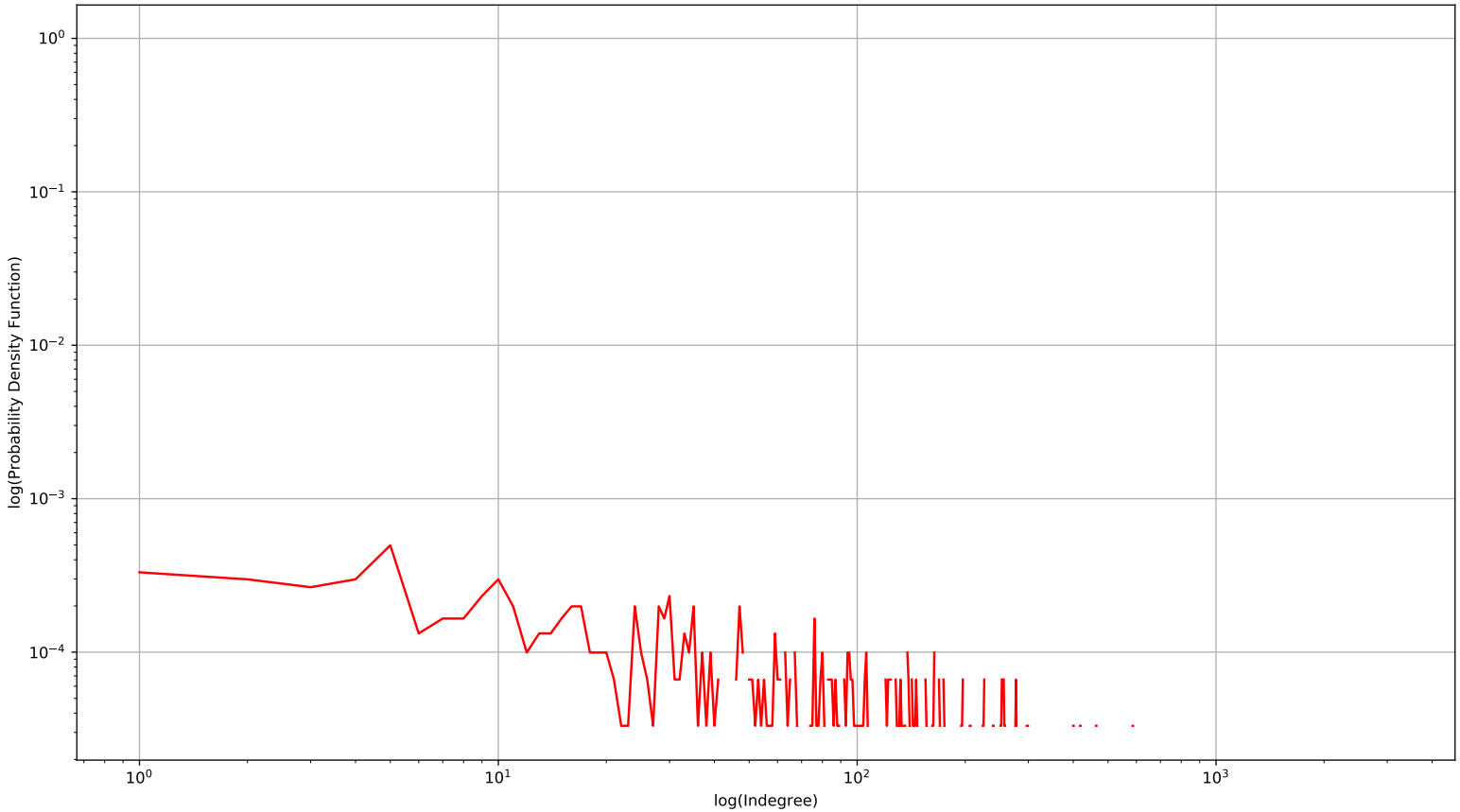
[2] La densità sensibilmente bassa è data dalla modalità di acquisizione dei dati.

Grado in entrata (1)

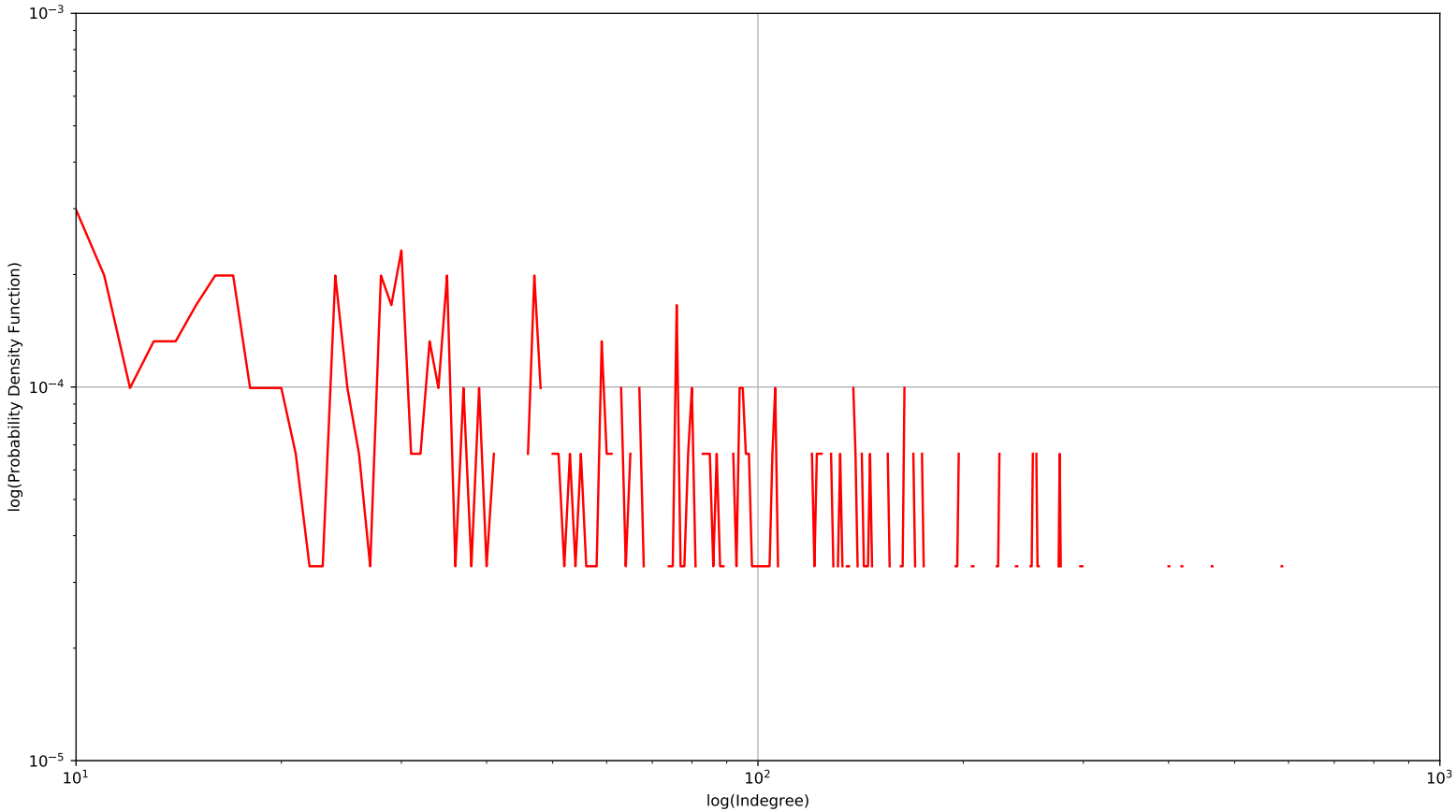




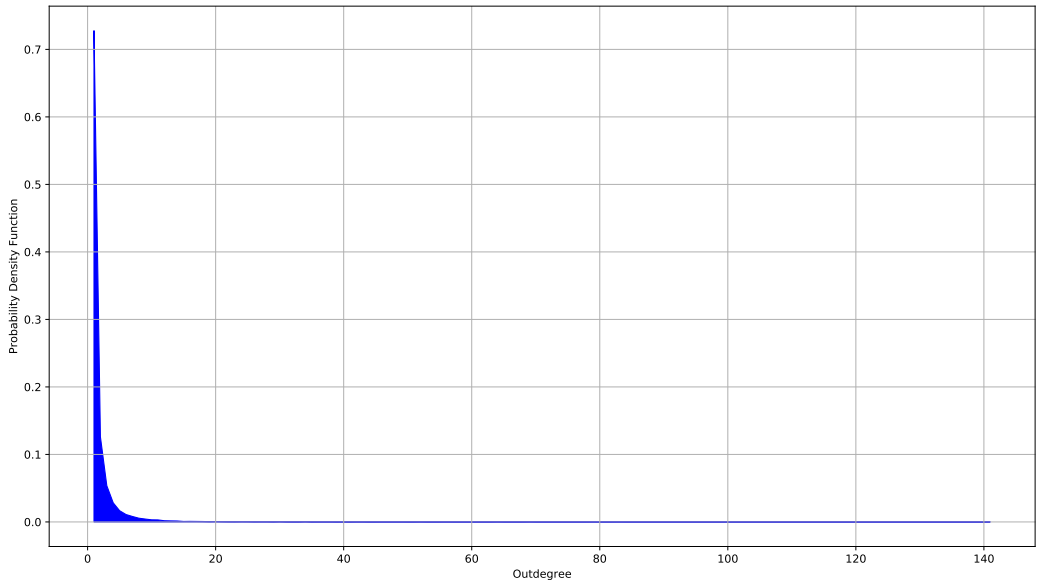
Grado in entrata (2)

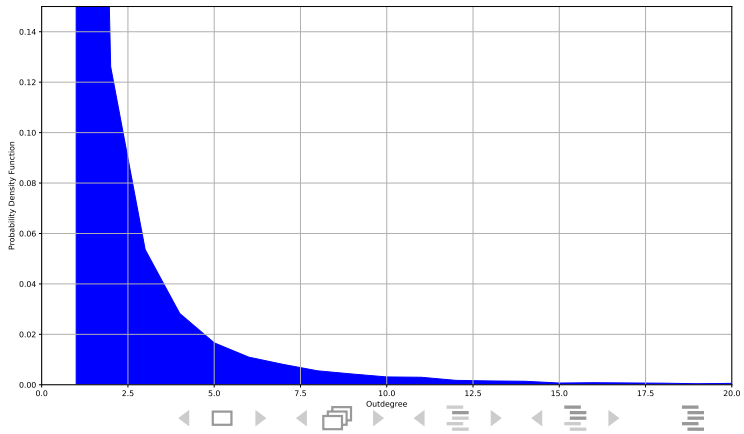


Grado in entrata (3)

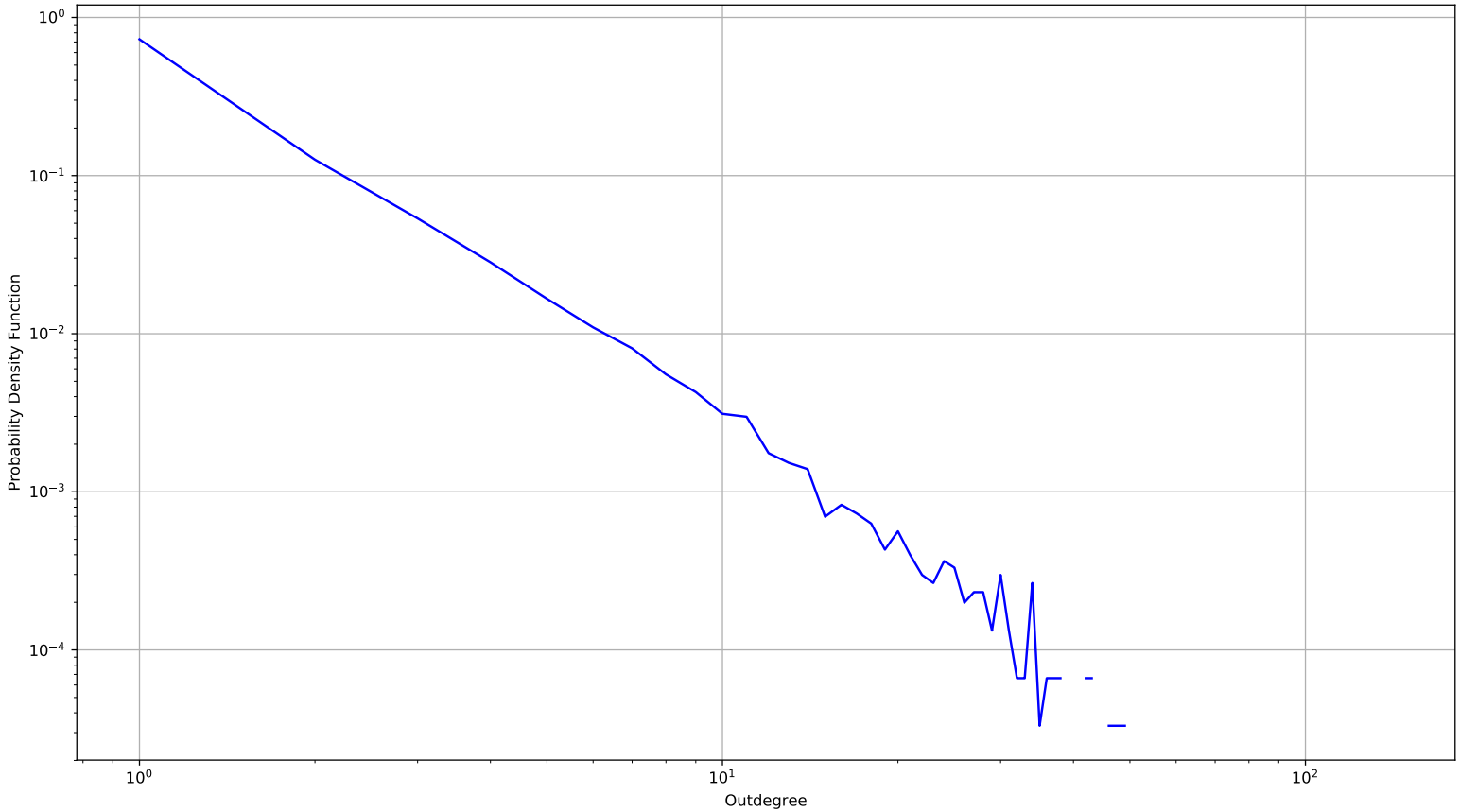


Grado in uscita (1)

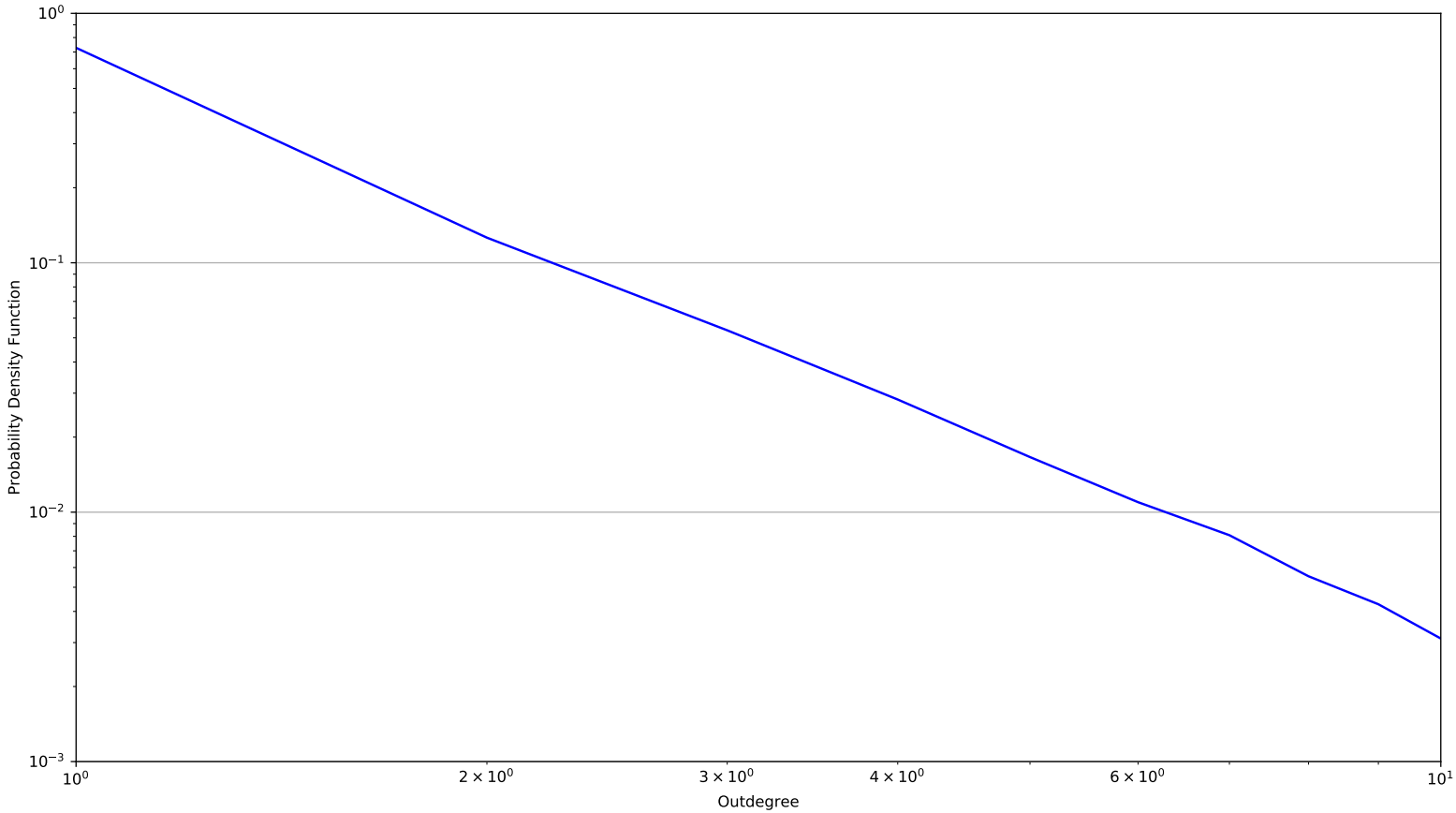




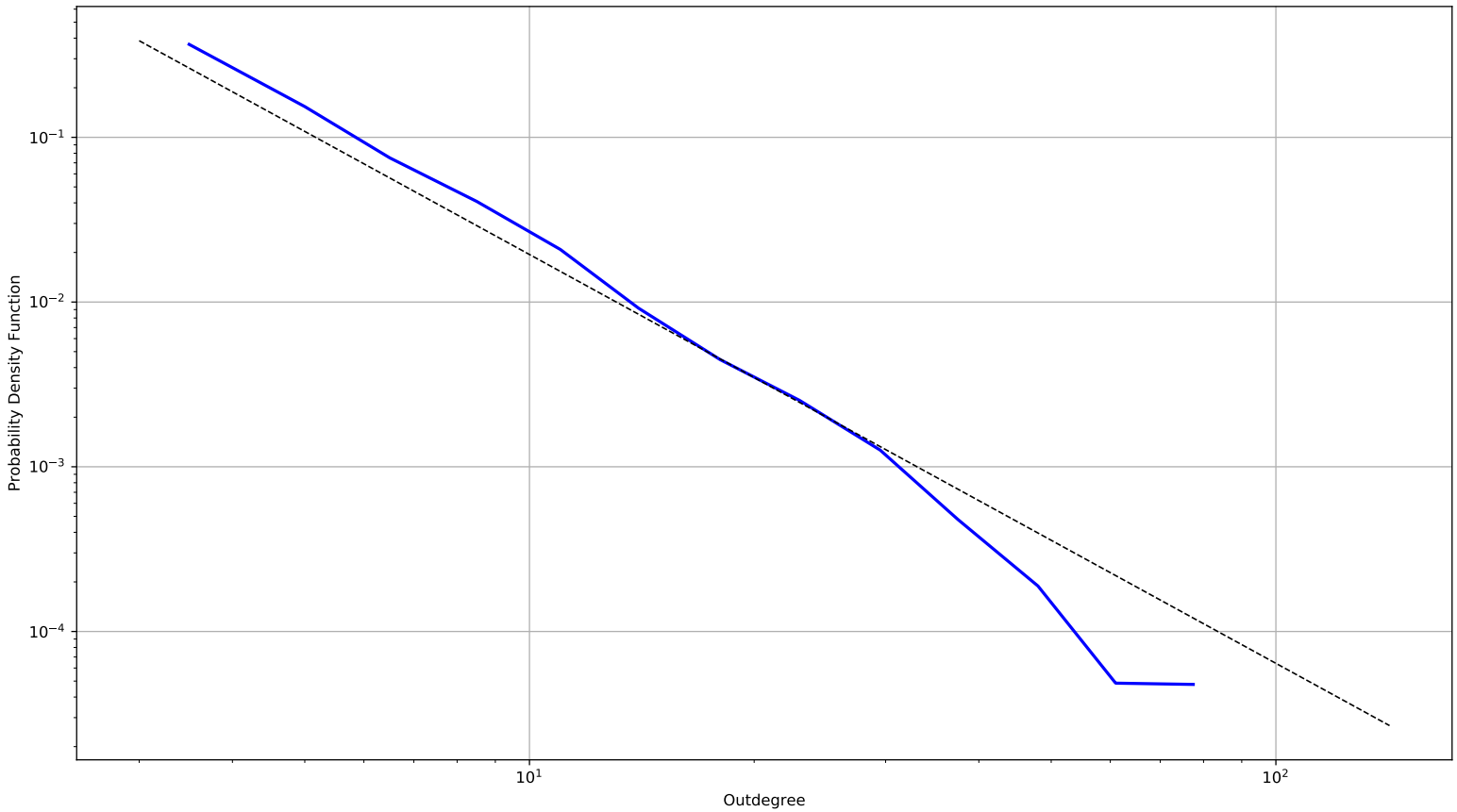
Grado in uscita (2)



Grado in uscita (3)

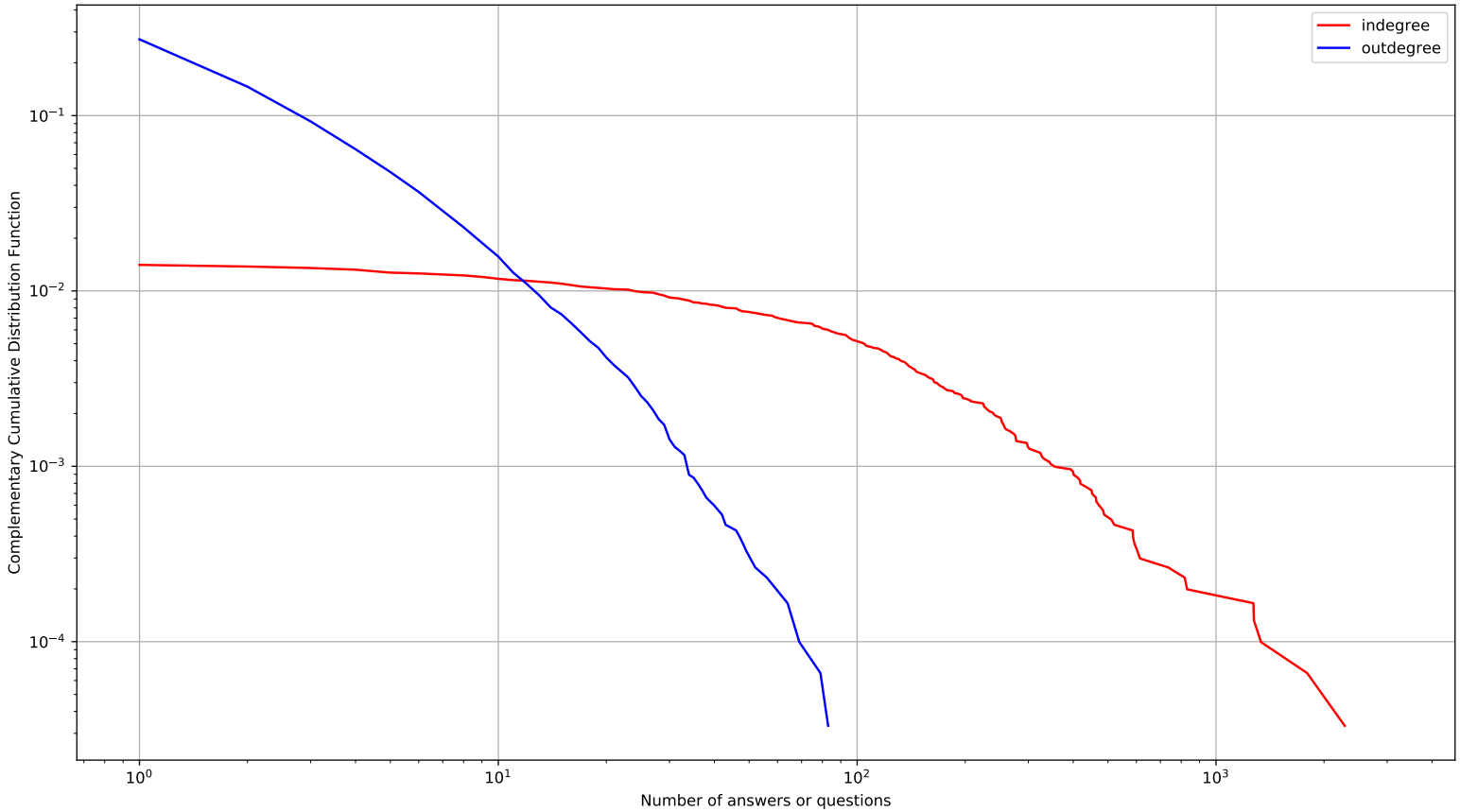


Adattamento della distribuzione ad una legge di potenza



L'esponente della legge di potenza è 2.482.

Confronto: indegree vs. outdegree



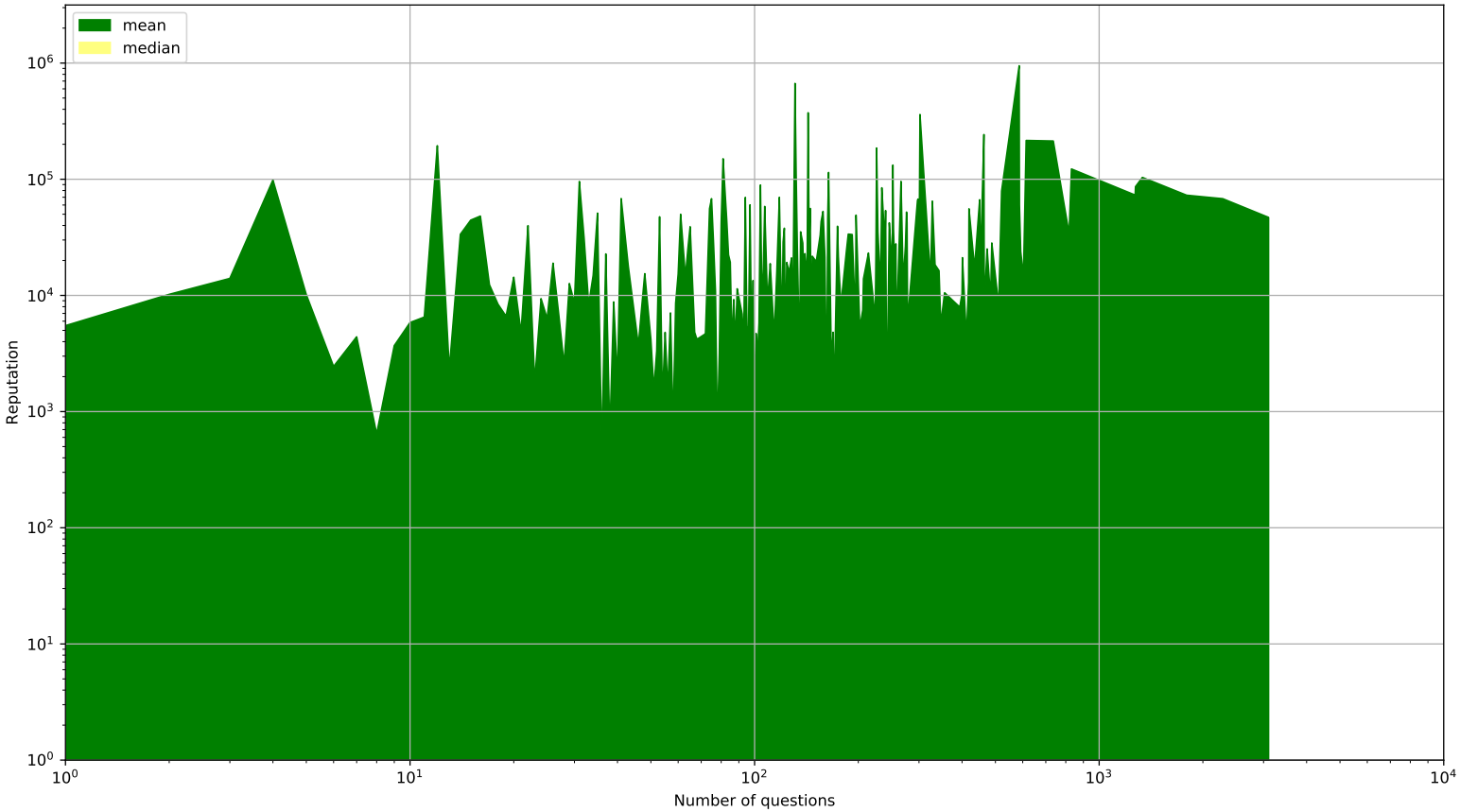
Indice

Reputazione

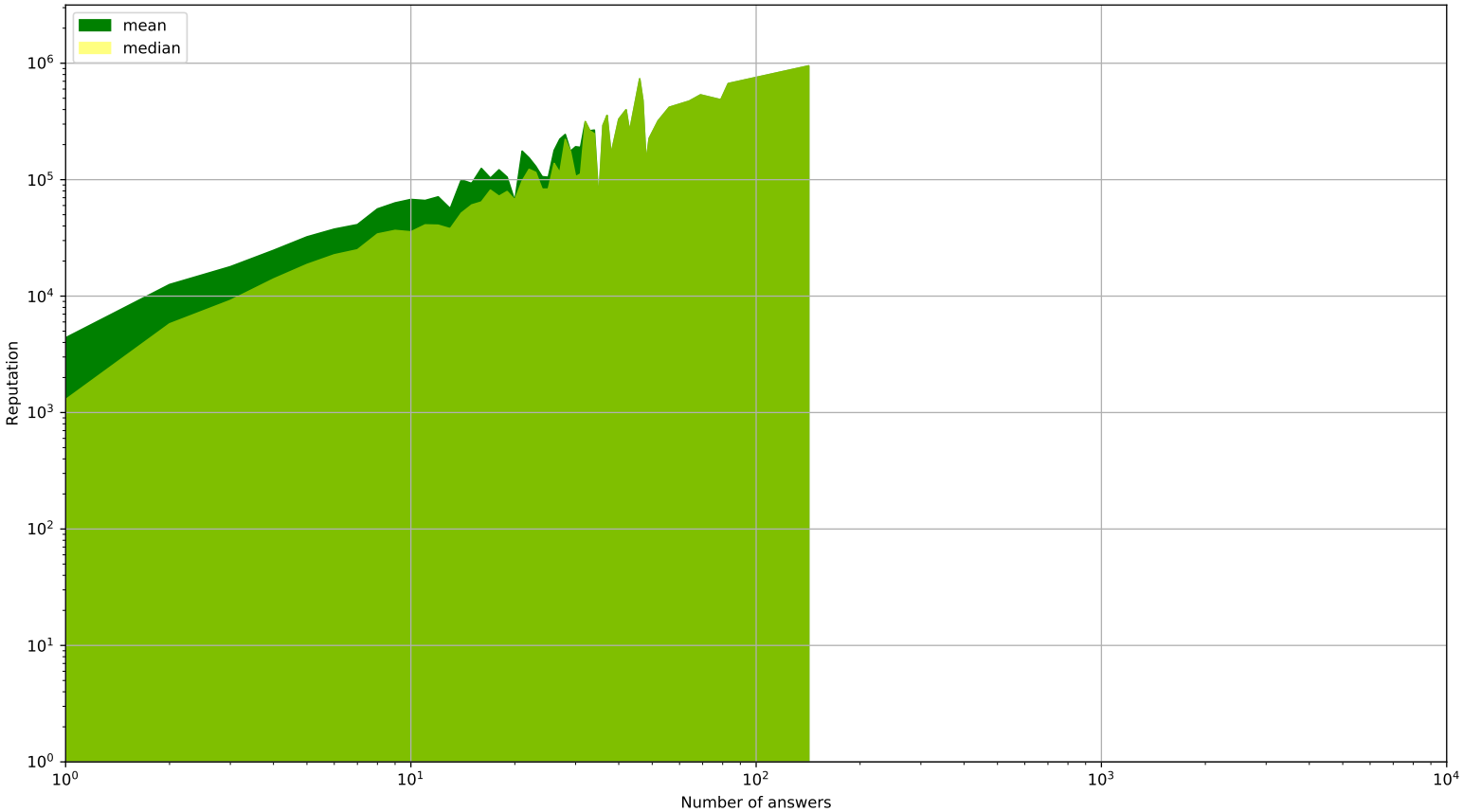
Ogni volta che un utente pone una domanda, pubblica una risposta o esegue altri tipi di azioni, gli viene attribuito (o tolto) un punteggio, la cui somma prende il nome di reputazione. Per quanto riguarda domande e risposte, i punteggi sono attribuiti così:

- +5: la domanda viene votata positivamente
- +10: la risposta viene votata positivamente
- +15: la risposta viene marcata come "accettata"
- -2: la domanda viene votata negativamente
- -2: la risposta viene votata negativamente
- -1: ogni volta che l'utente vota una risposta negativamente

Confronto: domande e reputazione



Confronto: risposte e reputazione



Indice

Ranking in esame

Si è deciso di confrontare tre misure di ranking diverse, una propria della rete studiata, una spettrale e un'ultima geometrica. Sono:

- Ranking per reputazione
- PageRank
- Closeness

Ranking per reputazione

#	Reputation	Name	#	Reputation	Name
1	948 734	Jon Skeet	11	545 536	Gordon Linoff
2	734 862	Darin Dimitrov	12	540 294	T.J. Crowder
3	728 702	BalusC	13	532 979	Quentin
4	695 870	Hans Passant	14	529 026	paxdiablo
5	681 822	VonC	15	516 899	Alex Martelli
6	668 653	Marc Gravell	16	503 350	dasblinkenlight
7	645 280	CommonsWare	17	491 849	CMS
8	579 946	SLaks	18	487 389	marc_s
9	555 922	Martijn Pieters	19	485 756	Mark Byers
10	553 514	Greg Hewgill	20	485 482	JaredPar

PageRank^{TM}

#	Reputation	Name	#	Reputation	Name
1	948 734	Jon Skeet	11	17 278	Alex Baranosky
2	46 955	acidzombie24	12	34 053	Pacerier
3	8 138	Wayne Molina	13	14 575	Toran Billups
4	72 868	Joan Venge	14	5 623	agnieszka
5	68 007	Edward Tanguay	15	23 890	Agnel Kurian
6	10 160	martinus	16	16 318	johnc
7	73 080	flybywire	17	213 426	Brian R. Bondy
8	86 374	Jason Baker	18	122 600	OscarRyz
9	635	Simon P	19	52 207	James
10	103 695	Claudiu	20	13 117	David Koelle

Closeness ranking

#	Reputation	Name	#	Reputation	Name
1	948 734	Jon Skeet	11	141 288	John Saunders
2	668 653	Marc Gravell	12	55 826	annakata
3	681 822	VonC	13	361 139	Johannes Schaub
4	695 870	Hans Passant	14	450 826	Eric Lippert
5	485 482	JaredPar	15	99 235	Brian
6	374 037	tvanfosson	16	45 566	JesperE
7	341 912	Konrad Rudolph	17	418 421	Reed Copsey
8	60 510	casperOne	18	242 707	Bill the Lizard
9	297 752	Mehrdad Afshari	19	215 773	Aaron Digulla
10	263 670	Joel Coehoorn	20	87 792	leppie

Indice di correlazione

Per avere un'idea della similarità dei tre metodi di ranking è stata calcolato questo indice di correlazione sui primi venti risultati:

$$\sigma_{U,V} = \frac{\mid \{x = y \mid x \in U, y \in V\} \mid}{\mid U \mid}$$

	Reputation	**PageRank**	**Closeness**
Reputation	100%	5%	10%
Pagerank	5%	100%	5%
Closeness	10%	5%	100%

τ di Kendall

Per effettuare una comparazione più rigorosa dei tre algoritmi di ranking è stata utilizzata l'approccio standard, calcolando la tau $\tau - b$ di Kendall, questo modo:

$$\tau_{U,V} = \frac{P - Q}{\sqrt{(P + Q + J) \times (P + Q + K)}}$$

Con P, il numero di coppie condordanti, Q, il numero di coppie discordanti, J, il numero di pareggi solo in U e K, il numero di pareggi solo in V.

	Reputation	**PageRank**	**Closeness**
Reputation	1	0.0616396	0.2027289
Pagerank	0.0616396	1	0.1688404
Closeness	0.2027289	0.1688404	1

Indice

Reprocità

La reciprocità è calcolata come il rapporto tra il numero di cicli di lunghezza 2 e il numero totale di archi nel grafo. Formalmente:

$$r = \frac{\mid \{(u,v) \in G \mid (v,u) \in G\} \mid}{\mid \{(u,v) \in G\} \mid}$$

Con $G = (V, E)$, grafo orientato.

Nel passaggio da grafo orientato a grafo non orientato è stato perso lo 0.31% di archi.
La reciprocità generale del grafo è 0.00625.
Per il nodo di partenza, Jon Skeet, la reciprocità è 0.39011.

Coefficiente di clustering globale

La transitività (o coefficiente di clustering globale) è la tendenza dei nodi di una triade a chiudere il triangolo ed è così calcolata:

$$T = 3 \times \frac{| \{(x,y),(y,z),(z,x) \in E \mid x,y,z \in V\} |}{| \{(x,y),(y,z) \in E \mid x,y \in V\} |}$$

Con $G = (V, E)$, grafo orientato.

L'indice di transitività della rete è 0.08427.