# Algorithm to determine thread "hotness"

Asked 16 years ago Modified 14 years, 8 months ago Viewed 2k times

2

I'm trying to come up with a way to determine how "hot" certain threads are in a forum. What criteria would you use and why? How would these come together to give a hotness score?

The criteria I'm thinking of include:

- how many replies
- how long since the last reply
- average time between replies

The problems this algorithm must solve:

- A thread which has 500 replies is clearly hot, unless the last reply was over a year ago.
- A thread with 500 replies that was replied to a second ago is clearly hot, unless it's taken 4 years to reach 500 replies.
- A thread with 15 replies in the last 4 minutes is really hot!

Any ideas, thoughts or complete solutions out there?

algorithm forum

Share

Improve this question

Follow

asked Dec 19, 2008 at 14:45

nickf

546k ● 198 ● 658 ● 725

this might be what you are looking for: stackoverflow.com/questions/32397/popularity-algorithm – kemiller2002 Dec 19, 2008 at 14:48

## 7 Answers

Sorted by: Highest score (default)

2

Jeff Atwood has a nice question about this with a ton of information on other "hot" algorithms. I suggest using one of those and adapting it to your liking.

Share Improve this answer

Follow

edited Mar 20, 2017 at 10:29

Community Bot

1 ● 1

answered Dec 19, 2008 at 14:51

EndangeredMassa

17.5k ● 8 ● 56 ● 80

1

Simplest algorithm: If there have been greater than X replies since Y, it is hot.

If you prefer something that scales, just count how many replies since time y. More replies means more hotness.

Share Improve this answer

Follow

answered Dec 19, 2008 at 14:49

Brian

**25.8k** ● 18 ● 86 ● 178

---

0

I was thinking you could probably model it with diminishing waves here, using amplitude (or root mean square) to measure hotness. As time goes, the wave diminishes, and so a late reply will only cause a little stir.

In practice, I think this requires a lot of calculation. You could make good use of caching to speed up the calculation.

Just my two cents.

Share Improve this answer

Follow

answered Dec 19, 2008 at 14:48

csl

**11.3k** ● 5 ● 63 ● 91

---

0

In short I've found logarithmic decay of "hotness" to be the most natural.

Share Improve this answer

Follow

answered Dec 19, 2008 at 14:50

Robert Gould

**69.7k** ● 61 ● 191 ● 275

0

Thanks to those who posted the links to the other questions/answers. Unfortunately, those equations take a lot more things into consideration than what's possible with my setup (eg: voting, reputation of author, etc)

After playing around with it, I've come up with this equation which I'll use for the time being:

```
log10($numOfReplies * 20000 /
pow($timeSinceLastPost, 1.3))
```

It still could use some work. For example, if there's a really really popular but old thread, it'll be low on the hotness, but if one person replies to it that'll put it right back to the top for a few days/weeks.

Share Improve this answer

Follow

answered Dec 19, 2008 at 16:16


nickf

546k ● 198 ● 658 ● 725

---

0

Why not just use a sort of exponential decay model. Hotness of thread = sum( k^(time since posting) ) for all posts. This has the advantage of being really easy to update and calculate. You'd have to play around with k and your unit of time measurement (k should be < 1, but fairly close to it)

Current hotness = hotness at time of last post * k^(time since last post).
Hotness after new post = current hotness + 1

Share Improve this answer

Follow

answered Dec 19, 2008 at 16:33



Yuliy

17.7k ● 6 ● 43 ● 47

---

0

One thing you should pay some attention to is whether people might want to "game" the algorithm in order to make/keep their threads "hot". Actually, you can pretty much assume that they will.

The minimum you should do to discourage this is to only consider replies from different people.

Share Improve this answer

Follow

answered Apr 16, 2010 at 5:44

Michael Borgwardt

346k ● 80 ● 486 ● 722