How to rank a million images with a crowdsourced sort

Asked 16 years, 2 months ago Modified 8 years, 3 months ago

Viewed 16k times

87

I'd like to rank a collection of landscape images by making a game whereby site visitors can rate them, in order to find out which images people find the most appealing.

What would be a good method of doing that?

- **Hot-or-Not style**? I.e. show a single image, ask the user to rank it from 1-10. As I see it, this allows me to average the scores, and I would just need to ensure that I get an even distribution of votes across all the images. Fairly simple to implement.
- **Pick A-or-B**? I.e. show two images, ask user to pick the better one. This is appealing as there is no numerical ranking, it's just a comparison. But how would I implement it? My first thought was to do it as a quicksort, with the comparison operations being provided by humans, and once completed, simply repeat the sort ad-infinitum.

How would *you* do it?

If you need numbers, I'm talking about one million images, on a site with 20,000 daily visits. I'd imagine a small proportion might play the game, for the sake of argument, lets say I can generate 2,000 human sort operations a day! It's a non-profit website, and the terminally curious will find it through my profile :)

algorithm **sorting** **crowdsourcing**

Share

Improve this question

Follow

edited May 7, 2009 at 17:16

Svante

51.4k ● 11 ● 83 ● 124

asked Oct 2, 2008 at 22:09

Paul Dixon

300k ● 54 ● 314 ● 349

1 I wrote a toy application use GAE which does something like this: rank.appspot.com. It uses the concept of momentum for each item which I suspect degenerates into a variant of ELO, though I developed it independently. Would be happy to share the python src. – freespace Oct 5, 2008 at 6:45

@freespace I'd be interested to see the Python source for your algorithm. – akaihola Jan 10, 2009 at 3:18

Maybe, with this project, you should try to set up a neural network (just for fun, of course), and use the **Pick A-or-B** input to train the network. Maybe you the neural network will be able to pick the most beautiful one, after a lot of training. – Martijn Courteaux Mar 7, 2011 at 8:49

12 Answers Sorted by: Highest score (default)

102

+500

As others have said, ranking 1-10 does not work that well because people have different levels.

The problem with the **Pick A-or-B** method is that its not guaranteed for the system to be transitive (A can beat B, but B beats C, and C beats A). **Having nontransitive comparison operators breaks sorting algorithms**. With quicksort, against this example, the letters not chosen as the pivot will be incorrectly ranked against each other.

At any given time, you want an absolute ranking of all the pictures (even if some/all of them are tied). You also want your ranking not to change **unless someone votes**.

I would use the **Pick A-or-B (or tie)** method, but determine ranking similar to the Elo ratings system which is used for rankings in 2 player games (originally chess):

> The Elo player-rating system compares players' match records against their opponents' match records and determines the probability of the player winning the matchup. This probability factor determines how many points a players' rating goes up or down based on the results of each match. When a player defeats an opponent with a higher rating, the player's rating goes up more than if he or she defeated a player with a

> lower rating (since players should defeat opponents who have lower ratings).

The Elo System:

1. All new players start out with a base rating of **1600**
2. WinProbability = 1/(10^((Opponent's Current Rating–Player's Current Rating)/400) + 1)
3. ScoringPt = 1 point if they win the match, 0 if they lose, and 0.5 for a draw.
4. Player's New Rating = Player's Old Rating + (K-Value * (ScoringPt–Player's Win Probability))

Replace "players" with pictures and you have a simple way of adjusting both pictures' rating based on a formula. You can then perform a ranking using those numeric scores. (K-Value here is the "Level" of the tournament. It's 8-16 for small local tournaments and 24-32 for larger invitationals/regionals. You can just use a constant like 20).

With this method, you only need to keep one number for each picture which is a lot less memory intensive than keeping the individual ranks of each picture to each other picture.

EDIT: Added a little more meat based on comments.

edited Sep 23, 2014 at 20:12

endolith

26.7k ● 35 ● 135 ● 202

answered Oct 2, 2008 at 23:05

Laplie Anderson

6,429 ● 4 ● 35 ● 37

5 Transitivity doesn't matter at all. You just want to aggregate peoples' opinion and you would expect them to disagree on ranking. People are a noisy source of data and not consistent. – Owen Oct 2, 2008 at 23:10

5 my point is that if you have A > B > C > A, then simply using the ">" as a comparison is a problem since your sort will never finish (correctly) and you list will be in a constant state of flux even if no further people are voting. My answer provides a solution to this problem. – Laplie Anderson Oct 2, 2008 at 23:19

1 I'm marking this as the accepted answer as it picks the bones out of my suggestion to use quicksort and includes a nice illustration of Elo. – Paul Dixon Oct 3, 2008 at 6:39

6 The elo system is definitely the way to go for ranking the A/B method. However, you might as well use a better method than the incremental method above. Have a look at Bayeselo: remi.coulom.free.fr/Bayesian-Elo – Fantius Feb 28, 2011 at 4:31

2 @endolith, that was in response to OPs idea of storing the comparisons and using quicksort on them. That doesn't work. Using comparisons to generate a score, then sorting the score works. – Laplie Anderson Apr 23, 2021 at 18:58

42

Most naive approaches to the problem have some serious issues. The worst is how bash.org and qdb.us displays quotes - users can vote a quote up (+1) or down (-1), and the list of best quotes is sorted by the total net



score. This suffers from a horrible time bias - older quotes have accumulated huge numbers of positive votes via simple longevity even if they're only marginally humorous. This algorithm might make sense if jokes got funnier as they got older but - trust me - they don't.

There are various attempts to fix this - looking at the number of positive votes per time period, weighting more recent votes, implementing a decay system for older votes, calculating the ratio of positive to negative votes, etc. Most suffer from other flaws.

The best solution - I think - is the one that the websites The Funniest The Cutest, The Fairest, and Best Thing use - a modified Condorcet voting system:

> The system gives each one a number based on, out of the things that it has faced, what percentage of them it usually beats. So each one gets the percentage score NumberOfThingsIBeat / (NumberOfThingsIBeat + NumberOfThingsThatBeatMe). Also, things are barred from the top list until they've been compared to a reasonable percentage of the set.
>
> If there's a Condorcet winner in the set, this method will find it. Since that's unlikely, given the statistical nature, it finds the one that's the "closest" to being a Condorcet winner.

For more information on implementing such systems the Wikipedia page on Ranked Pairs should be helpful.

The algorithm requires people to compare two objects (your Pick-A-or-B option), but frankly, that's a good thing. I believe it's very well accepted in decision theory that humans are vastly better at comparing two objects than they are at abstract ranking. Millions of years of evolution make us good at picking the best apple off the tree, but terrible at deciding how closely the apple we picked hews to the true Platonic Form of appleness. (This is, by the way, why the Analytic Hierarchy Process is so nifty...but that's getting a bit off topic.)

One final point to make is that SO uses an algorithm to find the best answers which is very similar to bash.org's algorithm to find the best quote. It works well here, but fails terribly there - in large part because an old, highly rated, but now outdated answer here is likely to be edited. bash.org doesn't allow editing, and it's not clear how you'd even go about editing decade-old jokes about now-dated internet memes even if you could... In any case, my point is that the right algorithm usually depends on the details of your problem. :-)

edited Aug 6, 2014 at 17:02

endolith

26.7k ● 35 ● 135 ● 202

answered Oct 2, 2008 at 22:46

Cody Hatch

8,927 ● 6 ● 31 ● 36

Thanks for the reference to Condorcet voting systems, that line of enquiry let me to this useful wikipedia page en.wikipedia.org/wiki/Ranked_Pairs – Paul Dixon Oct 2, 2008 at 22:55

These sites said they were "broken" and have since been abandoned. I don't know if the algorithm was buggy or just the implementation. – endolith Aug 6, 2014 at 16:56

12

I know this question is quite old but I thought I'd contribute

I'd look at the TrueSkill system developed at Microsoft Research. It's like ELO but has a much faster convergence time (looks exponential compared to linear), so you get more out of each vote. It is, however, more complex mathematically.

http://en.wikipedia.org/wiki/TrueSkill

Share Improve this answer Follow

answered Dec 16, 2009 at 18:06

user233179

The concepts of TrueSkill offer a lot of possibilities to rank things based off "matches." Similiar concepts are used by Bing to serve up relevant ads. I wrote a lot about the details of TrueSkill at moserware.com/2010/03/computing-your-skill.html – Jeff Moser Mar 18, 2010 at 17:03

8

I don't like the **Hot-or-Not style**. Different people would pick different numbers even if they all liked the image exactly the same. Also I hate rating things out of 10, I never know which number to choose.

Pick A-or-B is much simpler and funner. You get to see two images, and comparisons are made between the images on the site.

Share Improve this answer

Follow

answered Oct 2, 2008 at 22:20

Paige Ruten

176k ● 37 ● 182 ● 199

5

These equations from Wikipedia makes it simpler/more effective to calculate Elo ratings, the algorithm for images A and B would be simple:

- Get Ne, mA, mB and ratings RA,RB from your database.
- Calculate KA ,KB, QA, QB by using the number of comparisons performed (Ne) and the number of times that image was compared (m) and current ratings :

$$K = 800/(Ne + m)$$

$$Q_A = 10^{R_A/400}$$

$$Q_B = 10^{R_B/400}$$

- Calculate EA and EB.

$$E_A = \frac{Q_A}{Q_A + Q_B}$$

$$E_B = \frac{Q_B}{Q_A + Q_B}$$

- Score the winner's S : the winner as 1, loser as 0, and if you have a draw as 0.5,
- Calculate the new ratings for both using: $R'_A = R_A + K(S_A - E_A)$.
- Update the new ratings RA,RB and counts mA,mB in the database.

Share Improve this answer

Follow

edited Feb 8, 2017 at 14:09

Community Bot

1 ● 1

answered Dec 24, 2008 at 13:42

Osama Al-Maadeed

5,695 ● 5 ● 31 ● 48

4

You may want to go with a combination.

First phase: Hot-or-not style (although I would go with a 3 option vote: Sucks, Meh/OK. Cool!)

Once you've sorted the set into the 3 buckets, then I would select two images from the same bucket and go with the "Which is nicer"

You could then use an English Soccer system of promotion and demotion to move the top few "Sucks" into the Meh/OK region, in order to refine the edge cases.

Share Improve this answer

Follow

answered Oct 2, 2008 at 22:14

Chris Cudmore

30.1k • 12 • 59 • 95

4

Ranking 1-10 won't work, everyone has different levels. Someone who always gives 3-7 ratings would have his rankings eclipsed by people who always give 1 or 10.

a-or-b is more workable.

Share Improve this answer

Follow

answered Oct 2, 2008 at 22:26

Bill K

62.8k • 18 • 112 • 158

I appreciate that, but I figured if I ensure each image gets an equal number of votes, it should average out. Trouble is, I think I'd need about 10 votes on each image, which based on the numbers above would take me 13 years. By which time I'd have another 5 million images :) – Paul Dixon Oct 2, 2008 at 22:29

1 Since people tend to either go with the average or high/low, if you decide to do that I suggest you reduce to 1-5 instead of 1-10. – Bill K Oct 2, 2008 at 23:09

Wow, I'm late in the game.

3

I like the ELO system very much so, but like Owen says it seems to me that you'd be slow building up any significant results.

I believe humans have much greater capacity than just comparing two images, but you want to keep interactions to the bare minimum.

So how about you show n images (n being any number you can visibly display on a screen, this may be 10, 20, 30 depending on user's preference maybe) and get them to pick which they think is best in that lot. Now back to ELO. You need to modify you ratings system, but keep the same spirit. You have in fact compared one image to n-1 others. So you do your ELO rating n-1 times, but you should divide the change of rating by n-1 to match (so that results with different values of n are coherent with one another).

You're done. You've now got the best of all worlds. A simple rating system working with many images in one click.

Share Improve this answer

Follow

answered Mar 5, 2011 at 5:26

asoundmove

1,322 ● 3 ● 14 ● 28

3

If you prefer using the Pick A or B strategy I would recommend this paper: http://research.microsoft.com/en-us/um/people/horvitz/crowd_pairwise.pdf



> Chen, X., Bennett, P. N., Collins-Thompson, K., & Horvitz, E. (2013, February). Pairwise ranking aggregation in a crowdsourced setting. In Proceedings of the sixth ACM international conference on Web search and data mining (pp. 193-202). ACM.

The paper tells about the *Crowd-BT* model which extends the famous Bradley-Terry pairwise comparison model into crowdsource setting. It also gives an adaptive learning algorithm to enhance the time and space efficiency of the model. You can find a Matlab implementation of the algorithm on Github (but I'm not sure if it works).

Share Improve this answer

Follow

answered May 17, 2015 at 3:01

idailylife

174 ● 2 ● 14

2



The defunct web site whatsbetter.com used an Elo style method. You can read about the method in their FAQ on the Internet Archive.

Share Improve this answer

Follow

edited Sep 23, 2014 at 20:16

endolith

26.7k ● 35 ● 135 ● 202

answered Dec 19, 2008 at 6:36

quidnunc

1

Pick A-or-B its the simplest and less prone to bias, however at each human interaction it gives you substantially less information. I think because of the bias reduction, Pick is superior and in the limit it provides you with the same information.

A very simple scoring scheme is to have a count for each picture. When someone gives a positive comparison increment the count, when someone gives a negative comparison, decrement the count.

Sorting a 1-million integer list is very quick and will take less than a second on a modern computer.

That said, the problem is rather ill-posed - It will take you 50 days to show each image only once.

I bet though you are more interested in the most highly ranked images? So, you probably want to bias your image retrieval by predicted rank - so you are more likely to show images that have already achieved a few positive comparisons. This way you will more quickly just start showing 'interesting' images.

Share Improve this answer Follow

edited Oct 2, 2008 at 22:55

answered Oct 2, 2008 at 22:27

Owen

2,819 ● 1 ● 15 ● 9

I can see the initial ranking with page views, which might help also. – Paul Dixon Oct 2, 2008 at 22:51

that should say "seed", not "see"! – Paul Dixon Oct 2, 2008 at 23:24

it could be "pick best out of 4" and then it counts as 3 pairwise rankings for each vote – endolith Aug 6, 2014 at 16:55

1

I like the quick-sort option but I'd make a few tweeks:

- Keep the "comparison" results in a DB and then average them.
- Get more than one comparison per view by giving the user 4-6 images and having them sort them.
- Select what images to display by running qsort and recording and trimming anything that you don't have enough data on. Then when you have enough items recorded, spit out a page.

The other fun option would be to use the crowd to teach a neural-net.

Share Improve this answer

Follow

answered May 7, 2009 at 17:04

BCS

78.3k ● 69 ● 194 ● 298