

A bulky academically written part to a technical whitepaper

by Jesse and James

September 28, 2018

Contents

I	Ranking system for a dApp store	2
1	Decentralized ranking systems in context	2
1.1	Motivation	2
1.2	Companies that use voting systems	3
1.3	Resources and literature	6
2	On voting	7
2.1	On existing voting systems	8
3	Desired voting properties	14
3.0.1	On exposure/money as reward	17
3.1	Resources and literature	18
4	economical perspective	19
5	how the properties apply for other companies and how steemit is bad etc	19
6	Propose an algorithm	20

Part I

Ranking system for a dApp store

(This part ought to be included in a document with several parts.)

1 Decentralized ranking systems in context

1.1 Motivation

In this paper, we advance the discussion of voting systems on distributed ledgers such as blockchains. The core benefit of such frameworks is that vote-casting and voting-result-computation is perfectly transparent and thus always audible. More concretely, we will not study the consensus protocols enabling the decentralized voting system themselves, but instead review and propose

- algorithms for poll evaluations resulting in a sorted list
- innovations made possible through the underlying audible framework.

The former can manifest as a mechanism for the evaluation of opinions invoked by the voters, which, taken together, result in an average ranking. One of our main motivations here is the design of a voting system for a dApps store. So while the subject of the poll algorithms discussed may be anything from politicians to fruits, we will consider the rating of software applications. The vote casting actors will thus be referred to as ‘users’ and correspond to whitelisted addresses on the decentralized ledger. The whitelisting itself doesn’t need to be the result of a decentralized process.

In this context, the most notable innovation that comes with audible polling is the verifiable claim for rewards based on “on-chain“ actions. For each poll or voting round, the final result can be put in context with the individual user’s vote and thus used to compute a transparent reward claim tied to the user address.

TODO: Note on exposure/money as reward

1.2 Companies that use voting systems

Exploration of other platform's (blockchain and centralized ones') voting sy

- * Lisk voting, earnlisk.com
- * augur "reporting": 50% ROI
- * Gnosis
- * reward voting 7.0
- * openbazaar
- * repu-coin
- * odem.io
- * riskbazaar
- * drep.org
- * stackexchange

Blockchain related:

* Lisk:

-)delegate proof of stake --> one earns lisk by voting for delegates who share t
 -)4 batches max. 33 votes (max. 101 votes at altogether) to participate;
 -)to participate at a batch, one has to pay 1 lisk, which has to be in the lisk-
 -)(Open question: what happens, if voted delegators don't win --> is the paid li
- (source: <https://earnlisk.com/>)

Retailer related:

* Amazon:

Ranking factors: (<https://startupbros.com/rank-amazon/>)

*Conversion Rate Factors:

- Sales Rank
- Customer Reviews
- Answered Questions <--- add as factor to platform highlight algorithm
- Image Size & Quality
- Price
- Parent-Child Products
- Time on Page & Bounce Rate
- Product Listing Completeness

* Relevancy Factors

- Title
- Features / Bullet Points
- Product Description

- Brand & Manufacturer Part #
- Specifications
- Category & Sub-Category
- Search terms
- Source Keyword

* Customer Satisfaction & Retention Factors

- Negative Seller Feedback
- Order Processing Speed
- In-Stock Rate
- Perfect Order Percentage (POP)
- Order Defect Rate (ODR)
- Exit Rate
- Packaging Options

* ebay:

+) evaluation of sellers (quite simple):

standard evaluation, given by verified buyers:

- positive vote: + 1 point
- neutral vote: 0 points
- negative vote: -1 point
- one vote per buyer per week (Mon- Sun) is counted
- 13 different levels of rating of the sellers, symbolized by differently coloured stars

detailed evaluation, may be given after the standard evaluation:

- 1-5 stars (voting points) for each of 4 categories (article, communication, seller, shipping)
- independent from the standard evaluation, doesn't affect it
- one rating per purchase possible
- is are shown only if there are at least 10 detailed evaluations

+) evaluation of buyers (unimportant for nOS-purposes)

- buyers can be evaluated by the sellers, but only positive votes are possible

evaluations can be edited if both parties do agree

+) evaluation of products:

- 1-5 stars (5 being the best)
- in addition, there are 3 product-specific questions to answer (yes/no)
- the average of the stars-rating and the percentage of positive answers to the questions
- also, people can write reviews; reviews can be given a positive or negative vote

(sources: <https://pages.ebay.de/help/feedback/howitworks.html>, <https://verkaeufe>

Q&A platform related:

xxx

- * StackExchange
- * Quora
- * Reddit
- * News papers
- * ...

Gaming:

*) VotingPlugin

(some plugin for Minecraft)

allows one to give his players rewards by voting for his servers;

types of rewards:

-) for votes for one site
-) for voting on all of some specified sites
-) for the first vote
-) cumulative reward (vote x amount of times to be rewarded (per day/week))
-) for voting x number of times in a row
-) for x amount of global votes

(source: <https://www.spigotmc.org/resources/votingplugin.15358/>)

TODO:

- * for whitepaper: make USER EXPERIENCE section

1.3 Resources and literature

The mathematics of voting and elections: A hands-on approach.

Why voting dApps isn't voting parties

The "Handbook of Electoral System Choice" (by Josep M. Colomer, Georgetown Univer

Furthermore, we don't have winners as such, but instead obtain exposure.

"Electorama" Wiki

This wiki is designed to give "election-minded" people an overview of different

* Main page

<https://wiki.electorama.com>

* All pages:

<https://wiki.electorama.com/wiki/Special:AllPages>

"Electoral Knowledge Network" website

This website provides information and customised advice on electoral processes.

The "Administration and Cost of Elections" (short: "ACE") - Projects promotes el

Besides other information, the website contains global statistics and data and a

<http://www.aceproject.org/>

<http://aceproject.org/ace-en>

Wikipedia:

TODO

2 On voting

We want to tap into the rich pool of established knowledge on voting systems, to inform our options and guide our decisions in designing suitable algorithms. In this section we recall the basic established definitions and properties and then reflect upon our needs.

[fontsize=\small]

- 3) We're first going to give an overview of
 - * properties of voting systems (viewed on their own)
 - * (differentiation between) popular existing voting systems.

[fontsize=]

2.1 On existing voting systems

One way to measure the fairness of a given voting system is to determine some properties that the system should satisfy. Commonly, the three properties that a voting system should fulfil in order to be fair are anonymity, neutrality and monotonicity. For two-candidate elections they are defined as follows:

Anonymity: A voting system is said to be anonymous if it treats all voters equally. I.e. if any two voters trade ballots, this shouldn't change the election's outcome.

Concerning the dApp-voting-system, this property could be argued. For example, it could be sensible to give users who have a high reputation, which indicates their knowledge, or users who hold a large stake and therefore are likely to want the best for the platform, more voting power than others. Certainly it would establish an unwanted great inequality between users if the relation between voting power and reputation or stake was a linear one. We discuss this topic further in

Neutrality: A voting system is said to be neutral if it treats all candidates equally. I.e. if every voter switched their vote from one candidate to another, the outcome should change accordingly.

Monotonicity: A voting system is said to be monotone if it is impossible for a candidate to change from winning to losing by gaining additional votes and to change from losing to winning by losing votes without gaining others.

The so-called May's Theorem states that majority rule is the only voting system for two-candidate-elections that is anonymous, neutral, and monotone, and that avoids the possibility of ties. However, majority rule is no option for the dApp-voting-system because there will be more than two candidates and anonymity is no property wanted.

Some generalisation of majority rule, which majority rule is a special case of, is plurality method. It is the voting system that elects the candidate who receives the largest number of votes, even if that number is less than half of the total number of votes cast. Plurality method only results in a tie, when two or more candidates receive the same number of votes and more than the number of votes received by any of the other candidates.

Concerning the dApp-voting-system, the possibility of ties is no downside since there's no need to determine a single winner.

Rather, what is needed is a system that leads to some sort of preference order of all dApps. Such preference order produced by the voting system is called "societal preference order" since it can be thought of as the ranking of the candidates that, according to the voting system being used, best reflects the

voters' will.

There are various system that can be used to determine the societal preference order. One property of such systems that might sound sensible at first, is the following:

A voting system is said to satisfy the majority criterion if whenever a candidate is ranked first by a majority of the voters, that candidate will also be ranked first in the corresponding societal preference order. We give an example for why this would be no legitimate property for the dApp-voting-system in a moment.

Some voting system that does not fulfil the majority criterion, is the so-called Borda count which uses a point system to determine overall rankings and is often used in collegiate sports polls for example. In an election with n candidates it works as follows:

Firstly, each voter submits a ballot that contains his or her individual preference order of all the candidates.

Then points are awarded to each candidate for each ballot cast, according to the following rule:

A m -place ranking is worth $n - m$ points (where $1 \leq m \leq n$). In other words, a first-place ranking is worth $n - 1$ points, a second-place ranking is worth $n - 2$ points and so on. Finally the candidate whose total number of points from all of the ballots is the largest is declared the winner and the corresponding societal preference order is determined by the number of points each candidate has got from largest to smallest. If there is more than one candidate with the largest number of points, a tie occurs. Also some sort of tie occurs in the societal preference order whenever candidates receive the same number of total points. They then occupy consecutive indistinguishable positions in the preference order.

Intuitively, Borda count also seems to be quite fair. However, it violates the majority criterion. For the purpose of the dApp-voting-system we consider using some sort of variation of the Borda count more appropriate than sticking to majority criterion. Imagine an extreme situation where a small majority of voters (i.e. users) ranks a specific dApp first place but all of the other voters rank it last on their personal preference. Obviously it would make no sense to declare this dApp the winner and rank it first in a societal preference order. Determining it's rank according to the Borda count seems much more legitimate. So the dApp-voting-system will not fulfil the majority criterion and will not use the plurality method, but rather use some sort of Borda count in order to lead to a sensible societal preference order.

Taking account of personal and societal preference orders, for elections with more than two candidates, the three properties of fair voting systems have to be redefined:

Anonymity: A voting system is said to be anonymous if it treats all voters equally. I.e. if any two voters traded their personal preference orders, the outcome of the resulting societal preference order should not change.

As mentioned before, this is no property we want the dApp-voting-system to fulfil, see

Neutrality: A voting system is said to be neutral if it treats all candidates equally. I.e. if every voter switched the positions of two specific candidates on their personal preference orders, the positions of these two candidates in the resulting societal preference order would be switched accordingly.

Monotonicity: A voting system is said to be monotone if it is impossible for a candidate to go from winning to losing or to experience a decrease in rank on the resulting societal preference order whenever changes in favour of that candidate, but no changes in disadvantage of that candidate, occur on individual preference ballots.

Some properties one obviously would want a voting system to fulfill are the following:

1. A candidate who would defeat every other candidate in a head-to-head election (under majority rule) is called a Condorcet winner.
2. If a candidate would lose to every other candidate in a head-to-head contest (under majority rule) is called a Condorcet loser.
3. A voting system is said to satisfy the Condorcet winner criterion (CWC), if it always elects the Condorcet winner whenever one exists.
4. A voting system is said to satisfy the Condorcet loser criterion (CLC), if it always elects the Condorcet loser whenever one exists.

Since the definition of a Condorcet winner involves two-candidate-elections, for a voting system with more than two candidates it is a bit tricky to design it in a way that it fulfills the Condorcet criterion. The Borda count, for example, doesn't fulfill the criterion since it violates majority rule.

Overview

Properties of voting systems

Voting system criteria

https://wiki.electorama.com/wiki/Category:Voting_system_criteria

Features for classifications of voting systems

- * Plurality voting
- * Instant-runoff voting

Popular voting systems

First-past-the-post voting/Winner takes it all

- Voters indicate on a ballot the candidate of their choice, and the candidate w
- If there are at least two positions to be filled, each voter casts (up to) th
- (huge downside: it very much encourages tactical voting)
- https://en.wikipedia.org/wiki/First-past-the-post_voting

Majority judgement

- Used to determine a single winner.
- Voters grade each candidate in one of several ranks, for instance named from "
- The system's inventors mathematically proved that the system was the most "str
- The algorithm we propose is also based on the median grade.

https://en.wikipedia.org/wiki/Majority_judgment

Approval voting

- Usually used to determine a single winner.
- Each voter may select any number of candidates. The winner is the candidate wh
- Variation: each voter may only select a predetermined number of candidates, ot
- The algorithm we propose also allows each candidate to evaluate an arbitrary n

<https://www.electology.org/approval-voting>

https://de.wikipedia.org/wiki/Wahl_durch_Zustimmung

Closed list voting

- Used when positions are to be filled by members of the running parties.
 - Closed list voting: The voters only can vote for the parties. The elected part
- In praxis, the order in which a party's list candidates get elected can also be

- These systems aren't of any use for our considerations.
- https://en.wikipedia.org/wiki/Closed_list; <https://en.wikipedia.org/wiki/Party-list>

Preferential voting/Preference voting:

May refer to different election systems or groups of election systems. Some authors use the term "Preferential voting" may, for example, refer to ranked voting methods/ordinal voting, instant-runoff voting, range voting/score voting, open list, bucklin voting

Score voting/rate voting

- Used to determine a single winner.
- Voters rate candidates on a scale. The candidate with the highest rating wins.
- Variations of score voting can use a score-style ballot to elect multiple candidates.

TODOTODOTODO: comparison to majority judgement, approval voting and bucklin voting

Instant-runoff voting

Used in single-seat elections with more than two candidates
 Voters rank all of the candidates in their personal order of preference
 The candidate with the fewest first-choice-votes is eliminated. If there is more than one candidate with the fewest first-choice-votes, a runoff election is held between those candidates.
 In the last voting round there are only two candidates left. The one who gets a majority of votes wins.
 Variations: There are a few variations of Instant-runoff voting. For example, a "two-round" system where the candidate with the most votes in the first round is eliminated, and a runoff is held between the two remaining candidates.
 Downside for our purpose: well known (old) dApps are far too hard to be taken over

Open list voting

Voters have at least some influence on the order in which a party's candidates are elected.
 Open list describes a certain family of voting systems for elections in which multiple candidates are elected from a single party's list.
 In "relatively closed" list systems, a candidate must get a full quota of votes to be elected.
 In "more open" list systems, that quota is so low that it's possible that more than one candidate from the same party can be elected.
 In the "most open" list system, the total number of votes each candidate has received is used to determine the order in which they are elected.
 In a "free list" system/panachage electors even have more power over which candidates are elected.
https://en.wikipedia.org/wiki/Open_list; <https://en.wikipedia.org/wiki/Party-list>

Bucklin voting/The Grand Junction System:

- Usually used to determine a single winner.
- Each voter ranks the candidates in ascending order, the first one being the favorite.

- To evaluate the votings, at first the prime rank is considered. Each candidate
- The number of votes each candidate received at the second rank is added to the

https://en.wikipedia.org/wiki/Bucklin_voting
<https://de.wikipedia.org/wiki/Bucklin-Wahl>
<https://www.youtube.com/watch?v=CkIYZsJAvNQ>

Ranked voting/Ordinal voting systems

Ranked voting describes certain voting systems in which voters rank outcomes in

cardinal voting systems todo

https://de.wikipedia.org/wiki/Majority_Judgment
<https://www.electology.org/score-voting>
https://en.wikipedia.org/wiki/Score_voting

https://en.wikipedia.org/wiki/Instant-runoff_voting (todo)
https://en.wikipedia.org/wiki/Open_list
https://en.wikipedia.org/wiki/Preferential_voting (todo)

https://en.wikipedia.org/wiki/Ranked_voting (todo)

3 Desired voting properties

We now want to reflect upon the notions discussed in the preceeding section

..

(todo: this might be a bit intermingled with the above one atm.)

4) Then rule out and classify for the means of a) and b)

5) Also point out how those apply for other companies. TODO

Basic summary

Considerations we want and what we don't want

The voting system should fulfill the following criteria:

- 1) User's values: Reputation, Voting Power (reputation and voting power should be)
- 2) Reputation grows if user's vote is in consensus with community's votes
 - * a) various ways to predefine what "consensus" does mean
 - idea: determine consensus in periods of a week
 - * b)!!! but users should not be able to vote for apps they haven't used only
 - be in consensus
 - idea: reward could be higher for the first voters
 - idea: votes should cost
- 3) Reputation decreases if user's vote isn't in consensus with community's votes
- 4) bad apps may be reported by users (= some sort of downvoting in very, very bad)
- 5) the reputation of users should sink whenever their programmed apps get reported
- 6) if an app has a large number of reports (limit to be predetermined), users should
- 7) apps' values: number of votes, number of reports;
 - * evaluation based on:
 - a) usage rate (!!! but apps that are needed more often shouldn't have an advantage)
 - b) number of votes
 - c) number of reports

- d) time that an app has been in the market
- e) extent of NOS user base

8) user's reward for a consensus vote should not be dependent from user's voting

9) reputation and voting power might be limited

- a) idea: including a parameter so that at a very high level the increment

10) data onchain/offchain?

11) calculation costs

12) for a listed ranking, a dApp should only be added to the average if it has a

?) Rewards for dApp producers? (If so, it should probably not depend on ranking.

#####

Classification

Terminology and fundamental notions

! differentiate between voting systems and properties of such.

want <Ranking>

Majority (=absolute majority) => Plurality

but want people to not cast only one vote (otherwise we can too few votes).

how to count votes? Especially since there are several votes per person, i.e. up

|users| * |dApps|

votes

*) Ranking

will argue for a form of

https://en.wikipedia.org/wiki/Cardinal_voting

TODO: look at all of those and work out pros and cons and differences.

*) Anonymity, Neutrality, Monotonicity

Most fundamental base voting system is majority rule characterized by:

- Anonymity (Hodge p.4)
- Neutrality
- Monotonicity

Facebook likes

- Monotonicity is almost self-evidently valuable.
 - Neutrality is a free market rule. (While nOS has power over the system and can
 - Anonymity ... (see Steemit and Anonymity (reputation))
- => aggregation problem
=> solution via brackets of last time period
exposure effect

Regarding Steemit ... keep an eye on rewards: How it's solved by steemit, complex

*) Quotas

This is just details:

In our case,

1) Quotas <=> Could in principle be used as a cap for when a dApp even enters the
On the lower end (i.e. having a low quota) => Could be unfair w.r.t. exposure di
One could imagine a quota at the higher end (extra bonus exposure for breaking a b

2) Given a computed ranking, quotas <=> Could be used for a hard cap of which ap

Can be set for when typical numbers (user base, dApp base, size) are established
We have a ongoing running voting process with varying user base size and number

The majority of dApps should of course be accessible and searchable on the platform
=> keep an eye on rewards: Steemit system, complications

Both of the above => flagging

*) Required voting

Note that majority rule doesn't require everybody to vote and this is probably a
This (the lack of fixed number of total votes upfront) makes the design of the d

*) Ties

Due to the large numbers of dApps to be voted for and the high number of voting

*) Breaking Neutrality.

While it's not a crucial feature, a tie can easily be deterministically resolved

What makes Neutrality for us different than e.g. for political elections consider

It raises the question of how self-contained a voting round should be in general

One might be tempted to say it also frees us from incorporating previous data a

For our consideration

- * Reputation ...

- * Voting Power ...

- * Flag ... dApps can be annotated/singled out for being suspicious/spam

Established voting mechanisms

3.0.1 On exposure/money as reward

What does exposure mean?

- * a spot in a list (as opposed to relative quantitative gain, as in "Proportional")

Interested in all dApps => We want to use ranking

Positive votes (we vote who we want, not who we don't want). The ranking implicitly

For the users

- * Rewards

- * Reputation?

3.1 Resources and literature

- 4 economical perspective
- 5 how the properties apply for other companies and how steemit is bad etc

6 Propose an algorithm

dApp-producer ranking Algorithm details

Reward for both quality and quantity, and both should be necessary

Reviews

* Reward for writing reviews, judged on quality

Rewards

(note: The python file now contains a more elaborate variant of those ideas)

$S(n)$... stake of user n

$V(n)$... "how well" they voted

N =sum over n such that n voted right (that will be rewards) ... all users

M ... total reward

x ... fraction of non-stake-dependent rewards

$R(n) = (x * M) * f(v) + ((1-x) * M) * g(n, S(n))$

where $f(v)$ says how good they voted, sum over $f(v)$ is 1

and where $g(n, s)$ depends on the stake where $\sum_n g(n, s)$ is 1

On-chain requirements

todo: Write a summary of requirements when it comes to storage and script execution

tldr we need

* User accounts with a time by of on-chain data relating to past actions and current

* Readout of users ranking, computation of some numbers

* The result must definitely be on-chain, and if the write-back to the user accounts

TODO: explain the below in detail and put it in context

```
rankings = {
    'Alice':
        {
            'B+': [ 'cherry', 'orange' ],
            'D': [ 'banana', 'apple', 'kiwi' ]
        },
    'Bob':
```

```

{
    'B+': [ 'orange' ],
    'D+': [ 'kiwi ', 'pear ' ],
    'D': [ 'apple ', 'banana ' ]
},

'Carl': {
    'A+': [ 'pear ', 'apple ' ],
    'A': [ 'cherry ' ],
    'C+': [ 'peach ' ],
    'C': [ 'orange ', 'kiwi ' ],
    'F': [ 'banana ' ]
}
}

## collect data
grade_bases = to_rating_dict(GRADES)

fruit_user_ratings = {fruit: {} for fruit in FRUITS}
user_fruit_ratings = {user : {} for user in USERS }

for user in USERS:
    print( '\nuser: _{ }_'.format(user) + 40* '-' )
    user_ranking = rankings[user]
    for grade in GRADES:
        if grade in user_ranking.keys():
            gb = grade_bases[grade]
            l = user_ranking[grade]
            r = to_rating_dict(l)
            for fruit , step_rating in r.items():
                fr = gb-(1-step_rating)/len(GRADES)
                fruit_user_ratings[fruit][user] = fr
                user_fruit_ratings[user][fruit] = fr

## compute bracketed means
fruit_user_ratings_means = {fruit: mean(rating_dict.values()) for fruit in FRUITS}
fruit_user_ratings_means_sorted = sorted(fruit_user_ratings_means.items())

TODO: make the above code prettier

```