infoblox

WHITE PAPER

InfoRanks: Infoblox Ranking Service

Statistical inference for defining internet ranks

By: Laura Teixeira da Rocha

Table of Contents

Why we created InfoRanks

InfoRanks is Infoblox's product for generating statistically significant, accurate rankings of popular domains. InfoRanks uses statistical inference techniques that circumvent the effect of data variability and noise to provide insight into a domain's rank and its stability over time.

Rank lists of domains, IPs, and other criteria are widely used for security and internet applications. Infoblox, for example, has implemented a technique that relies on rank lists to determine whether a domain should or should not be allowed in its security products. Other organizations, such as security operation centers, use rank lists to verify the likelihood of a threat in their networks and to fill DNS resolver caches, which are used to improve performance.

Rank lists based on observation counts are vulnerable to the instability of internet data. For the remainder of this paper, we will focus on the application of domain rankings and on ranks determined by counting and sorting, although these techniques generalize to other methods for determining ranks.

The research that Infoblox and other organizations have conducted on publicly available rank lists, such as Amazon's Alexa Top 1 Million domains, has shown that ranks can vary considerably in just a few days. In particular, ranks based on observation counts in a network can be affected by incorrectly collected data, congestion in the network, seasonality, user trends, and other factors. For example, observation counts might decrease due to a network configuration that reduces traffic to a domain, or they might increase as a result of a marketing campaign that drives traffic toward the domain. Moreover, internet traffic generally follows the distribution model predicted by Zipf's law, which implies that as a domain loses popularity, small amounts of noise in the environment can have a major impact on the domain's rank.

This instability affects the performance of commercially available products and solutions that rely on such lists. In contrast, our ranking system for domains provides not only a range of plausible ranks but also the most likely single rank observed during a particular period. We use a statistical inference technique to create a statistically significant rank list that describes a domain's stability over time and gives us relatively high confidence in the most likely rank.

We combine data collected over a period of time and use that data to define a rank's confidence interval (CI) for each domain. With our method, a rank of a highly popular domain has a smaller CI and lower variation, which means that the rank is stable. We obtain these results by aggregating historical data and simulating its ranking with a bootstrap sampling technique. This approach generates repeated values that will be used to estimate the most likely rank. It also generates unbiased estimators by using a sampling technique with replacement, where each rank or value has the same probability of being selected in each round; this allows us to avoid selection bias.

We can define how confident we want to be about a domain's rank range. If we want to be highly confident, we can compute a 99% CI and create wider ranges for each rank; a 99% CI will generally have a strong statistical significance, 0.01%, because it incorporates more cases for the rank values. If we want to be less confident (that is, less strict) about the confidence level, we can choose a 90% CI, which will generate narrower ranges of ranks for a specific domain.

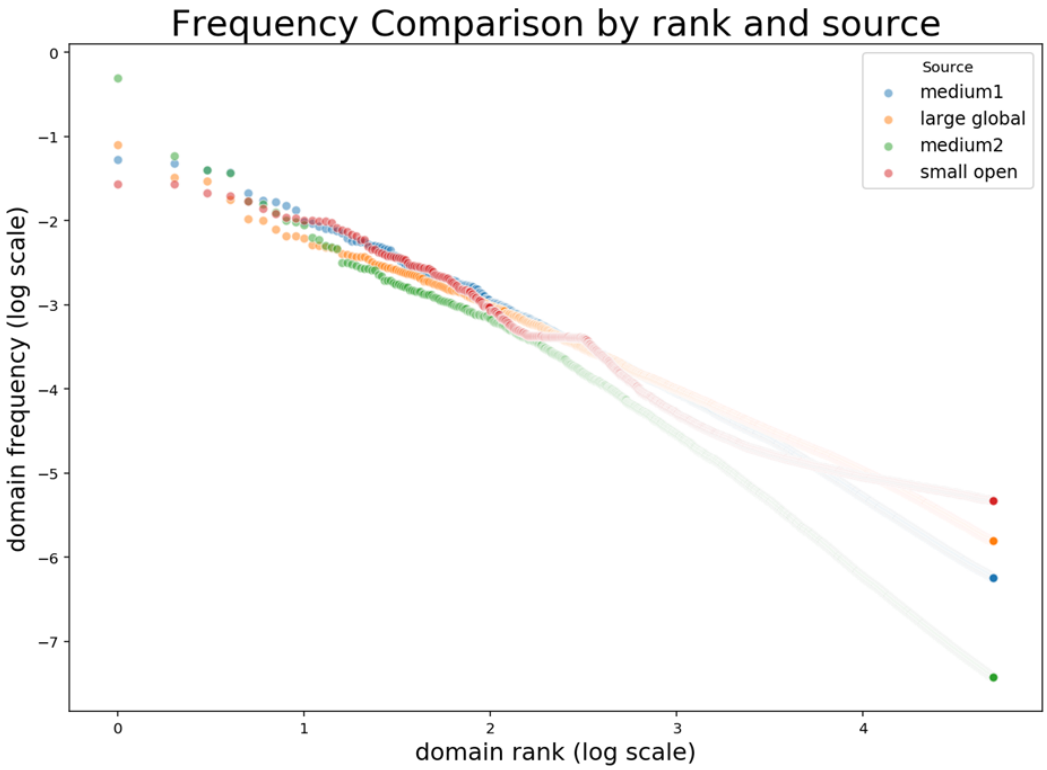
Using the bootstrap sampling technique allows us to meet the assumption that the data is normally distributed, and that the samples are selected randomly and are independent of each other. This data-driven approach is defensible because it allows us to specify how confident we want to be in the results, create ranks resistant to variability, and gain insight into the ranks' stability over time.

Sources of instability

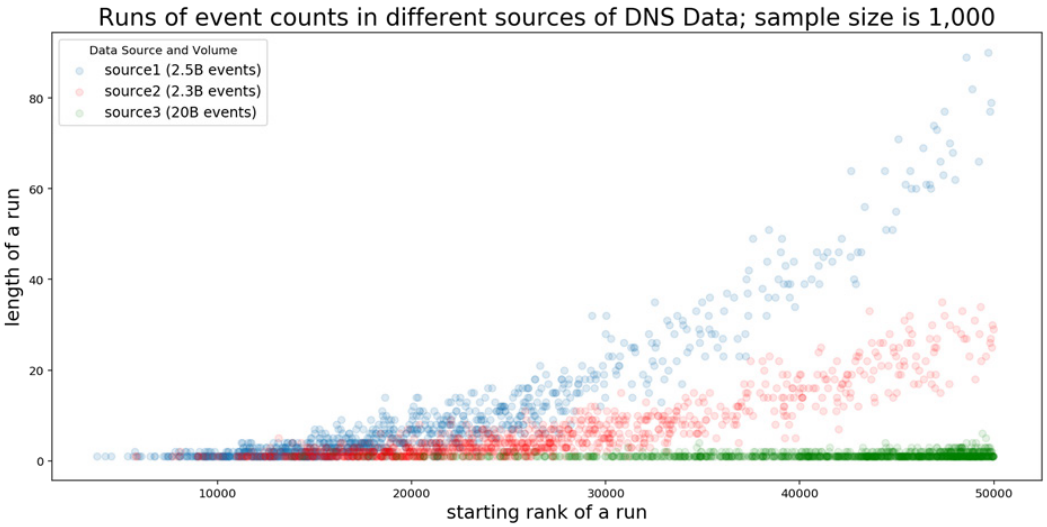
Suppose that the rank of a domain is defined as an index in a list of domains ordered by the number of daily DNS queries (observations) for each domain. The rank is likely to vary over several consecutive days, and the less popular the domain, the more the rank will vary. There are a number of sources of rank instability, some caused by the domain's traffic and some by external factors.

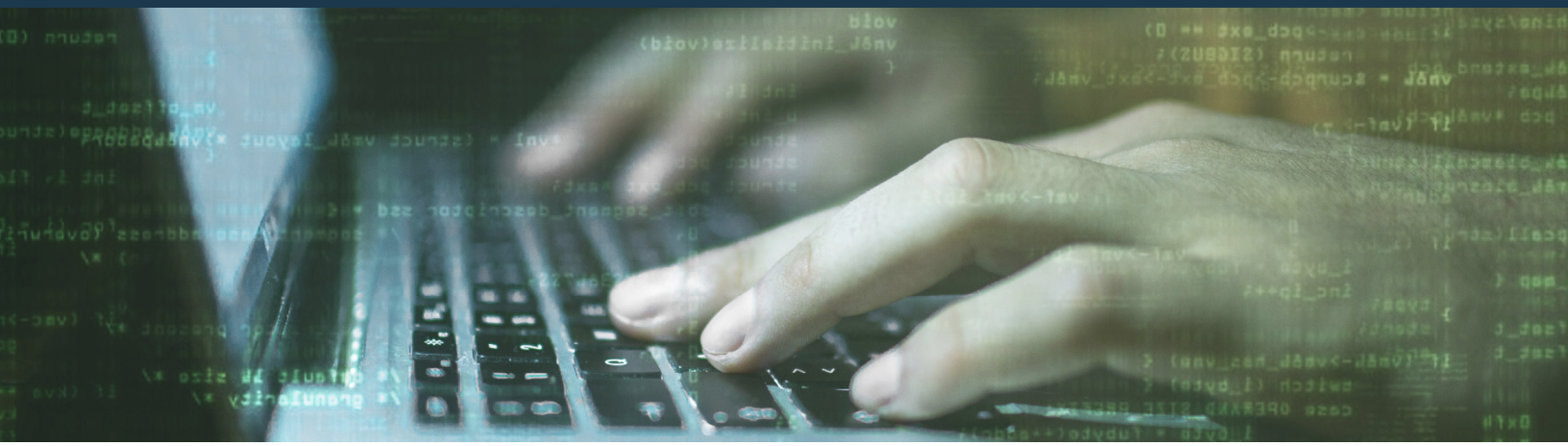
An unavoidable cause of a rank's instability is the natural distribution of DNS queries in a network. The frequency of DNS queries follows Zipf's law, which is an inverse discrete power law distribution. The probability density function can be modeled as $P(r) = Cr^{-\alpha}$, where α for internet domains is approximately 1.0 , C is a scaling constant, and $r \geq 0$. According to this equation, the most popular domain would have rank 0, and this would imply that the count between consecutive iterations for a less popular domain would decrease by progressively smaller numbers. This opens the rankings up to perturbation caused by lost packets and minor differences in traffic. Moreover, the smaller the collection apparatus, the greater the perturbation. Because this distribution is discrete, every value must be an integer. The constant C reflects the size of the observation pool; small values of C create a distribution that turns flat quickly.

The plot below shows the effect of Zipf's law on various sizes of collection apertures for DNS domain data. Each data point in the plot also shows that rank data likely follows a broken power law distribution, which means the value for α might change.



Another way to think of this phenomenon is in terms of the counts themselves and of the necessity to consider runs in the counts during the creation of rankings. In this context, a run is a series of domains that have the same observation count. Under Zipf's law, runs are inevitable: a frequency cannot continue to decrease from a fixed number of items without eventually repeating a count on consecutive ordered events. Again, the size of the collection aperture has a significant effect on the lengths of a run. Sources with a small collection of data are forced to soon have domains that are seen the same number of times, and this creates a run in their data. The diagram below compares this effect for systems of three sizes.





For the purpose of computing ranks, Zipf's law matters for multiple reasons. First, if we observe 100 domains the same number of times, what ranks do we assign to them? Traditional calculation assigns ranks in alphabetical order or randomly, and this immediately creates a variance of 100. Second, there are numerous sources of noise—on the internet in general, and in DNS specifically—including lost packets. With 1% packet loss, for example, possible fluctuations in domain counts might cause the ranks obtained through count-sort methods to vary widely over a few days. Domain counts in DNS are also affected by the records' TTL (time to live), which is subject to caching, administrators' actions, and other regular events.

A domain's rank is affected not only by variance in observation counts but also by external forces. For example, daily and even hourly spikes in traffic have been observed for websites for political campaigns during debates and voting days, for new as well as established websites during marketing campaigns, for websites of small newspapers during political scandals, and for sports websites during games. Malicious actors also cause variance in DNS traffic. For example, phishing domains typically have strong, short-lived spikes in traffic; malware command and control domains might operate over short periods or in cycles.

Finally, the environment in which the observation or collection of data is made can have a major impact on the accuracy and interpretation of ranks generated by simple counting. For example, data collected between authoritative name servers and recursive resolvers contains only cache misses from the recursive resolvers. The cache time, and whether the resolvers are configured to pre-fetch domains, will affect the count of observed queries. Similarly, data collected between a recursive resolver and clients, which might themselves be resolvers, contains queries that are not cache misses for the entire network.

Implementation of InfoRanks

Suppose that we have an environment where we observe DNS queries and that we can collect and count the domains at some regular interval, such as daily. These counts have noise due to natural and external factors (see above), and a ranking that is based on simple ordering is likely inaccurate. To compensate for these issues, we assume that the data contains noise that we can mitigate statistically over time; specifically, by making statistical measurements over several days, we can reduce the effect of outlier counts and converge on a statistically significant representative value and a range. This section describes that process.

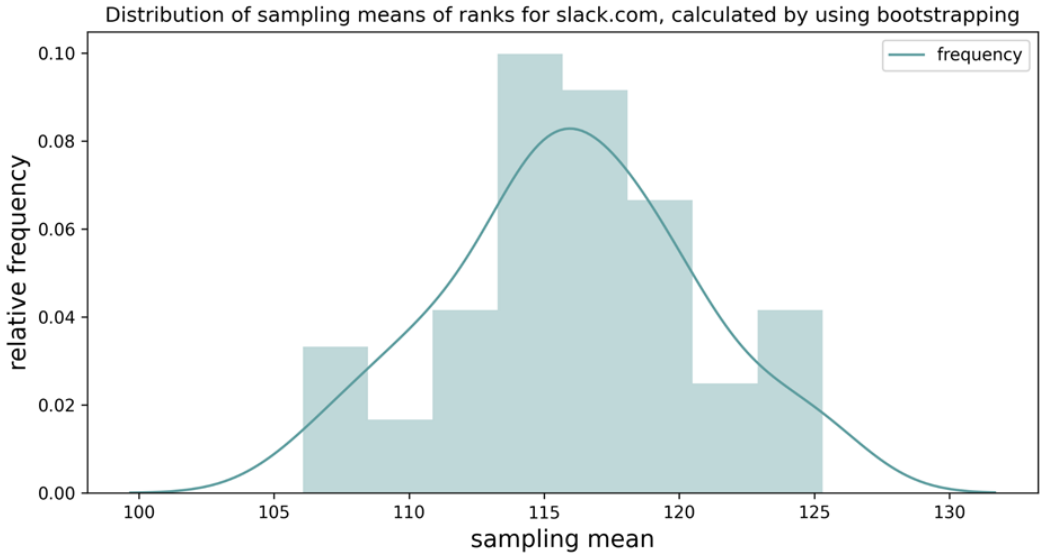
Given the population of D domains observed over a set period of time, T (for example, several days), we want to determine the CI for the rank of each domain: D_1,...., Dn. Each domain will have a daily rank — R_1,....,R_T — that is based on ordered observation counts or some other measure. If T is seven days, we have seven ranks for a specific domain, and we will use these ranks to compute our CI for each domain. Using $CI = Xbar \mp Z_{a/2} * s$, we obtain a CI's lower

and upper boundaries, defined here as $(r1,r2)$ or $(min\ rank,max\ rank)$. To compute a CI, we need our data to follow an approximately normal distribution, as per the assumption made for this method. To approximate a normal distribution for the daily T ranks associated with each domain, we bootstrap the samples for each domain to generate $S \geq 30$ repeated sampled ranks of size m each. We sample with replacement, to ensure that each rank has the same probability of being selected in each round. Using a randomized sampling technique to estimate the rank for each domain allows us to avoid having biased estimators for our ranks.

For each S_i daily rank sample of size m, for each D_i domain, we compute the statistic of interest: in this case, the sample mean. The central limit theorem proves that with a large enough sample (usually 30 or larger), the sampling distribution of the mean follows a normal distribution.

Estimating the most likely rank

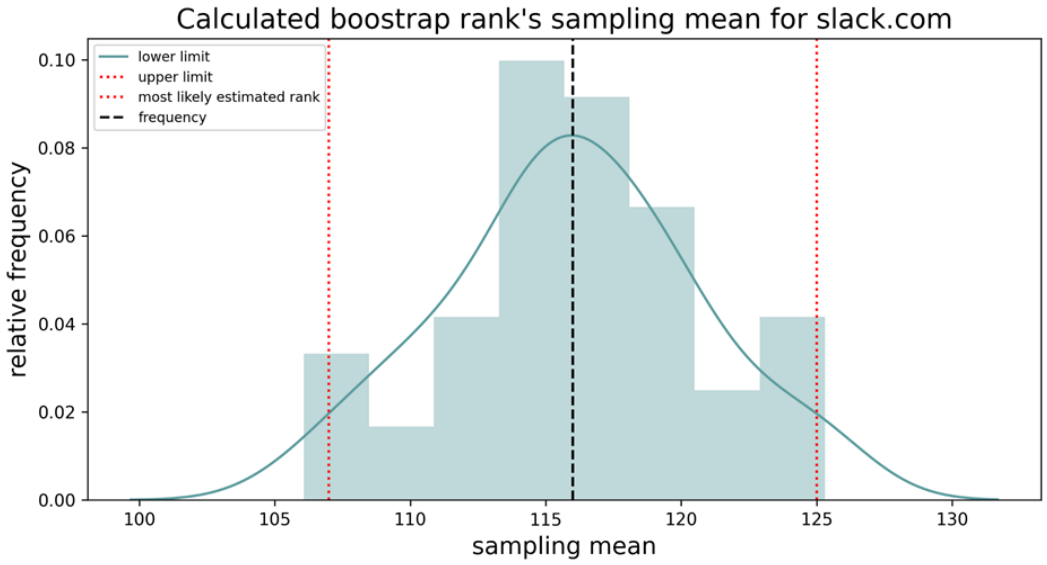
As an example, we will aggregate the daily ranks for domain example[.]com over seven days. The $R1,....,RT$ daily ranks associated with this domain during the seven days are 2426, 2576, 2426, 2576, 2576, 2521, and 2426. We take S repeated samples of size m, in this case 50 repeated samples of size 30 each, by using bootstrap sampling with replacement, and we compute the mean for the 30 sampled daily rank elements within each of these 50 repeated bootstraps. The mean sampling distribution for the domain shows that we have approximately met the normalization requirement:

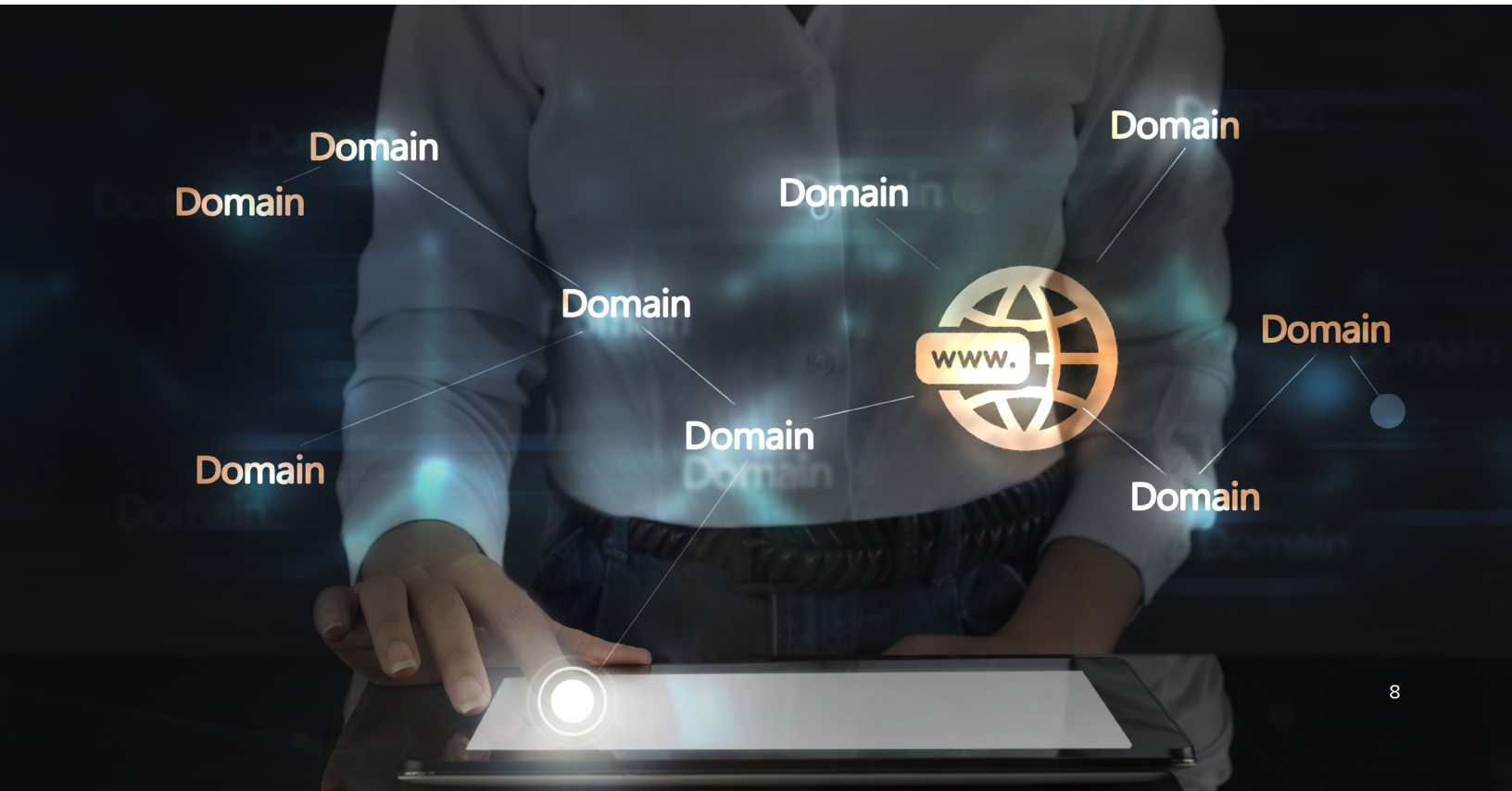


From the sampling distribution, we can obtain the maximum likelihood where the peak point is the highest probability of occurrence. We can use the maximum likelihood to infer the most likely true rank for the domain. Furthermore, we can determine the range of ranks that contains the most likely rank for the domain, without considering noise.

CIs for ranks

Using the sample statistic we have generated, and considering that the data is distributed normally, we can compute CIs for the normalized ranks of each domain. Recall that we now have the S-rank sampling statistic that we will use to obtain our CIs. With 95% confidence, we can confirm that 95% of the time, the S-rank statistic will fall between 2462 and 2535 for the seven days under consideration. With 99% confidence, the domain's rank will fall between r1 + x and r2 + x for 99% of the time, resulting in a larger CI range. The distribution's end tails that define the interval boundaries represent outliers and are least likely to appear due to the variance in the environment.

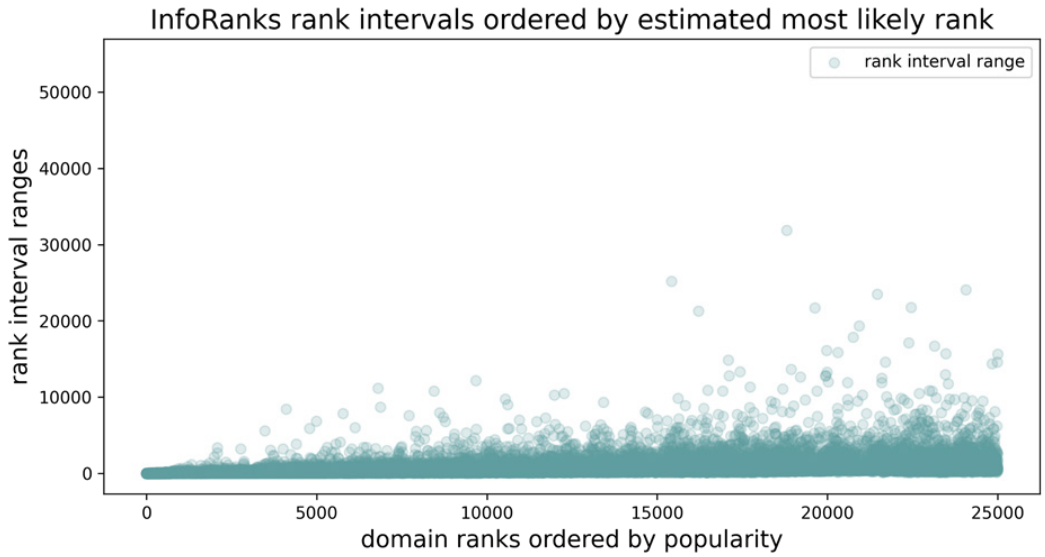




In this case, the CI range is computed by subtracting the lower boundary from the upper: r2 - r1. By following the same process for all D domains in the list, we can obtain CIs and CI ranges for each domain. The next section shows the results and describes how rank certainty decreases with a domain's popularity.

Analysis of variation in an interval range

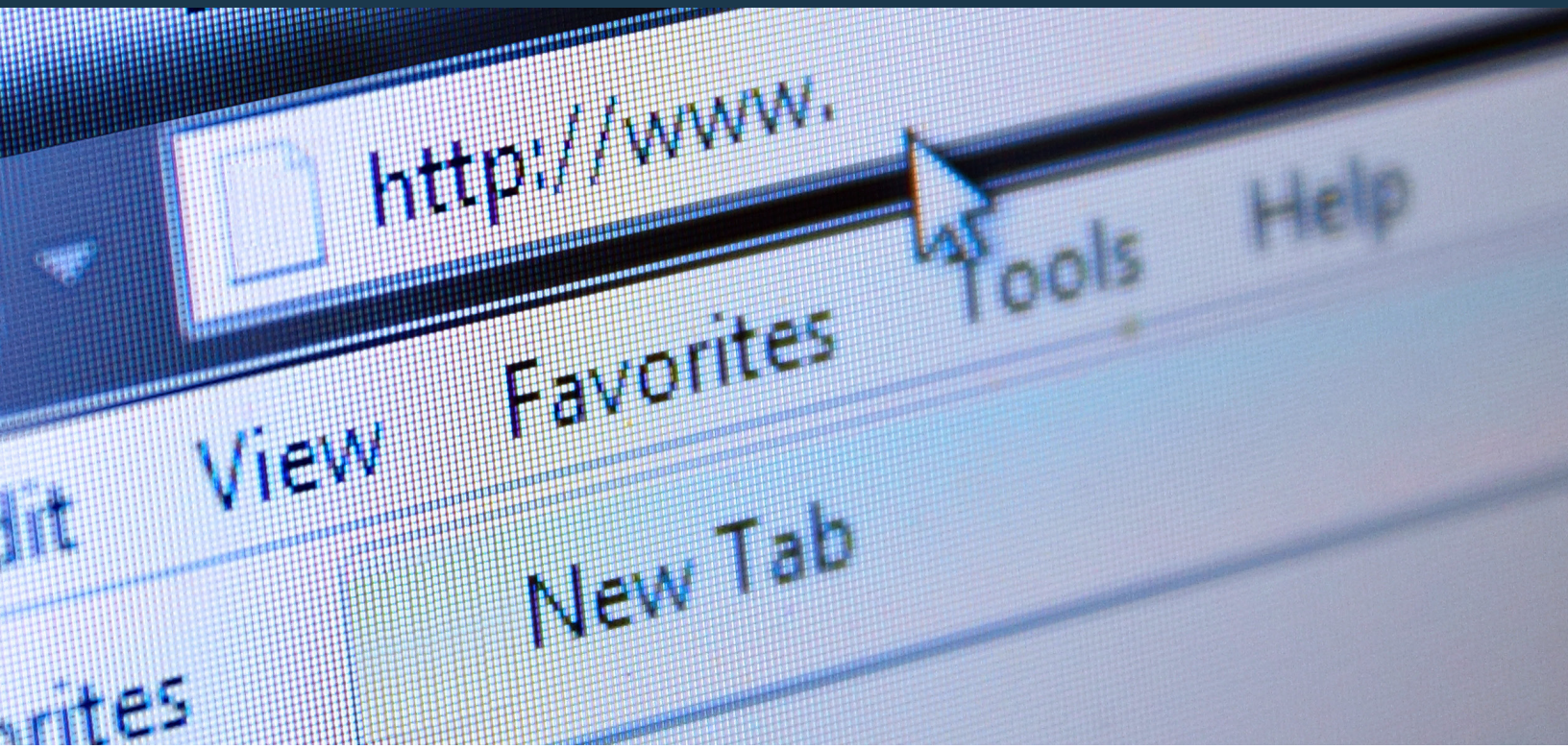
The rank intervals allow us to obtain the interval range metric, r2 - r1, as a way of summarizing the information about a rank's stability over time. The y-axis in the plot below shows the previously defined CI range, and the x-axis shows the domains ordered according to the most likely rank. The plot shows that as the domain loses popularity, the CI range widens. This occurs due to the direct relationship between a domain's popularity and variability: the less popular a domain (that is, the larger the rank values), the higher the rank's variability. Thus, ranks of unpopular domains have high variability.



Similarly, the more popular a domain, the smaller the CI range. This means that for highly popular domains, the ranks are more stable and the created CI ranges are narrower.

As an example, google[.]com has ranks across 7 days (3, 4, 4, 4, 2, 4, and 4), and this generates a very narrow range of (3, 4). A range that is only one rank wide is very stable over time and would increase our confidence in defining this rank. Generic confidence, which can be stable, somewhat stable, or unstable, is based on measurable "believability" metrics that reflect confidence and variability of a domain rank over time. Having representative data collected over time, and having insights into stability, enable users to filter data according to their use cases.

This approach allows domain ranks to collide: that is, to be identical for different domains. A common approach used for observations that have the same counts is to order the data according to a random aspect, such as lexicography, and assign different ranks even if the domains have the same counts and should thus have the same rank. Our method provides accurate information because it allows indicators (in this example, domains) to have the same ranks. This is similar to having two second-place winners in a sports competition.

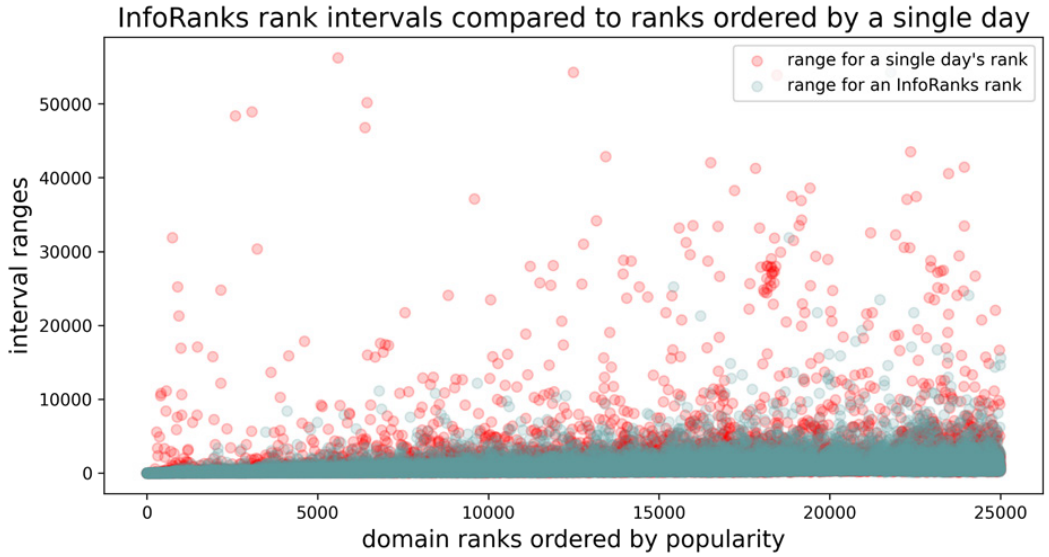


System analysis

With our approach, users can combine and select the data according to use cases. For example, to have domains with ranks that are highly stable over time, a user needs only to select domains whose ranks vary little. The rank ranges are used for providing insights about the stability of domains in a network.potential impact of a DDoS attack:

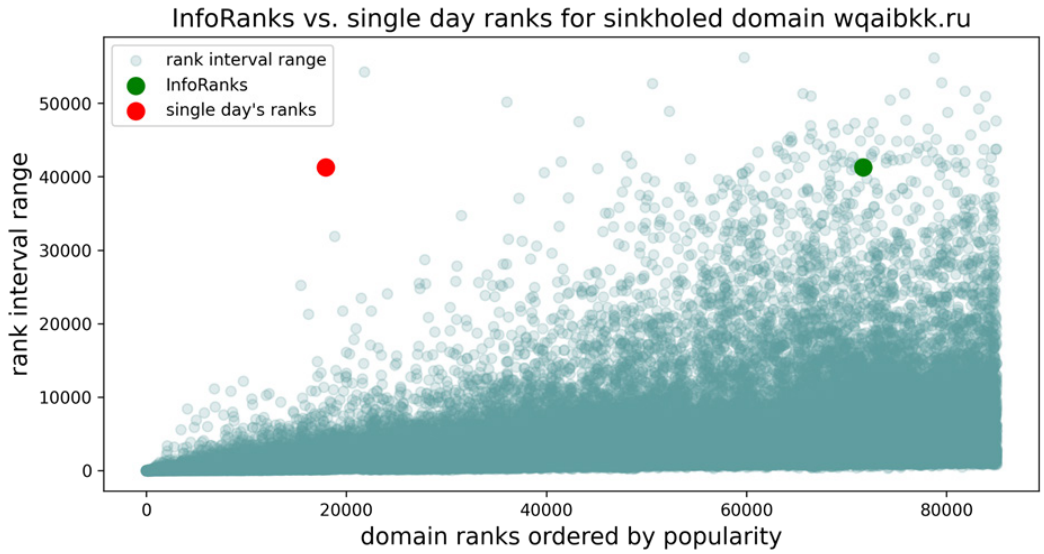
Example of InfoRanks vs. single ranks

This section illustrates the behavior and variability of estimated ranks over a period of time rather than a single day. The plot below shows that ranking domains by using a single day's worth of data makes domains with high variance in ranks appear more popular than stable domains.



Example of instability

Because ranks vary across days, an estimate based on a single rank is inaccurate. In contrast, rank intervals are not only accurate but also show a domain's instability over time; this information helps users (1) assess the amount of variation in the DNS environment and (2) make decisions confidently, because they are based on stable information. Our approach circumvents the effect of noise, defined as the occurrence of a high rank due to a spike in traffic on a single day. As demonstrated in the plot below, which shows a sinkholed domain ranked highly in popularity on a single day, our approach pushes the domain to a lower popularity due to the instability of its rank.



Our customers can access InfoRanks through our Customer Service Portal.

References

Batty, M. Rank Clocks | A Science of Cities. Accessed: August 9, 2019.

Becchetti L., Castillo C., 2006. The distribution of pageRank follows a power-law only for particular values of the damping factor. WWW '06: Proceedings of the 15th international conference on World Wide Web. Presented at the 15th international conference, ACM Press, Edinburgh, Scotland, p. 941.

Callahan T., Allman M., Rabinovich M., 2013. On modern DNS behavior and properties. ACM SIGCOMM Computer Communication Review, volume 43, issue 3, July 2013, pp 7–15.

CI*Rank - Confidence Intervals For Ranks of Cancer Incidence and Mortality Rates - Surveillance Research Program. The Division of Cancer Control and Population Sciences (DCCPS). March 25, 2011. Accessed: August 12, 2019.

Kendall, M.G., 1938. A New Measure of Rank Correlation. Biometrika 30, 81–93.

Mahanti A., Carlsson N., Mahanti A., Arlitt M., Williamson C., 2013. A tale of the tails: Power-laws in internet measurements. IEEE Network 27, 59–64.

Marshall E. C., Spiegelhalter D. J., Sanderson C., McKee M. Reliability of league tables of in vitro fertilisation clinics: retrospective analysis of live birth rates Commentary: How robust are rankings? The implications of confidence intervals. BMJ, vol. 316, no. 7146, pp 1701–1705, June 1998. BMJ 1998;316:1701.

Mohamad D. A., Goeman J. J., van Zwet E. W. An improvement of Tukey's HSD with application to ranking institutions. Cornell University. arXiv:1708.02428. November 2018.

Morrison J., Simon N. Rank Conditional Coverage and Confidence Intervals in High-Dimensional Problems. J Comput Graph Stat. 2018;27(3):648-656. doi: 10.1080/10618600.2017.1411270. Epub 2018 June 14. February 2017.

Newman, M.E.J., 2013. Power laws, Pareto distributions and Zipf's law. Cities 30, 59–67.

Newson R. Confidence Intervals for Rank Statistics: Percentile Slopes, Differences, and Ratios. The Stata Journal, vol. 6, no. 4, pp 497–520, November 2006.

Ogilvy H. Reinforcement learning assisted search ranking. Medium, January 11, 2018. Accessed: January 25, 2020.

O'Madadhain J., Hutchins J., Smyth P. Prediction and Ranking Algorithms for Event-based Network Data. ACM SIGKDD Explorations Newsletter, volume 7, issue 2, December 2005, pp 23–30.

Pochat V. L., Van Goethem T., Tajalizadehkhoob S., Korczyński M., Joosen W., 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. Proceedings 2019 Network and Distributed System Security Symposium.

Rweyemamu W., Lauinger T., Wilson C., Robertson W., Kirda E., 2019. Clustering and the Weekend Effect: Recommendations for the Use of Top Domain Lists in Security Research. In: Choffnes D., Barcellos M. (eds) Passive and Active Measurement. PAM 2019. Lecture Notes in Computer Science, vol 11419. Springer, Cham. https://doi.org/10.1007/978-3-030-15986-3_11.

Scheitle Q., Hohlfeld O., Gamba J., Jelten J., Zimmermann T., Strowes S.D., Vallina-Rodriguez N., 2018. A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists. IMC 2018: Proceedings of the Internet Measurement Conference 2018, October 2018, pp 478–493 https://doi.org/10.1145/3278532.3278574.

Wang Z., 2013. Analysis of DNS Cache Effects on Query Distribution. The Scientific World Journal, vol. 2013, Article ID 938418, 8 pages, 2013.

Wicklin R. Ranking with confidence: Part 1 - The DO Loop. SAS. Accessed: August 12, 2019.

Wood M. Bootstrapped Confidence Intervals as an Approach to Statistical Inference. Organizational Research Methods, vol. 8, no. 4, pp 454–470, October 2005.