Quantifying Criticality

Rob Pike

Introduction

We have a set of signals for a package that we would like to combine into a single number representing the *criticality* of that package. A package with higher criticality is one that is more important within its packaging system (NPM, RubyGems etc.) and therefore may deserve higher scrutiny and attention.

We start with ground rules. Assume that each signal is quantifiable, that is, that it can be represented by a number. Without loss of generality, specify that value must be ≥ 0. The goal is to find a single value that represents all the signals for a package in a meaningful way, possibly one that is evaluated differently in different systems. Since even within a single system, not all signals will be present for all packages, the mechanism must allow for that. Also, the set of signals is likely to expand over time, and possibly shrink.

The metric will contain arbitrary elements, such as the amount individual signals contribute to the criticality. There is a large element of judgment and refinement required in defining the precise way in which criticality is computed. Nonetheless, we seek a well-defined framework within which we can represent, refine, and justify such subjective adjustments.

Basics

Typical signals might be the number of downloads of a package or the number of dependents it has. For the purposes of this discussion, we will use the term *package* as the unit we are interested in, whatever it may be, and *signal* as an abstract, non-negative value that increases with importance but is otherwise not interpreted here.

We have a set of N signals $S_{pkg,i} \geq 0$. Not all packages need have values for all signals. For simplicity of notation we write just S_i for $S_{pkg,i}$ with the understanding we are talking about a particular single package.

Some signals are more important than others, and so with each signal S_i is associated an arbitrarily settable positive weight α_i that is consistent for all packages within the system. The weight α_i does not have a package-specific component.

Defining criticality

A naive starting point would be to define the *criticality value* of a package C_{pkg} as a weighted sum of the signal values for that package:

$$C_{pkg} = \sum_{i}^{N} \alpha_i S_i$$

This has several disadvantages. The value is unbounded, which means that the criticality value of a package may grow over time although the true importance of the package is unchanged. Also, many if not most signals are non-linear in effect, with Zipfian-like distributions. A package with 10,000 dependents is surely more important than a package with only 1,000, but arguably not ten times more.

Looking at the second point first, we can define for each signal S_i a function $f_i(S)$ that scales the signal non-linearly. In general we may need a different f_i for each i, but it is likely that a unique function will suffice, and that the logarithm is a reasonable choice. The precise function is unimportant to this analysis, though, and so for simplicity we will use the logarithm, adding 1 to avoid problems with having a zero for the signal value, and to make the minimum value 0:

$$C_{pkg} = \sum_{i}^{N} \alpha_i \, log\,(1 + S_i)$$

This has better scaling properties than the naive version. For one signal, the number of dependents, and a weight of 1, package A with 10,000 dependents has criticality value $C_A = 9.2$, while package B with 100,000 dependents has $C_B = 11.5$.

More important than the actual criticality value is how it compares to that of other packages. In the example from the previous paragraph, the ratio $C_B \div C_A$ is 1.25, which seems reasonable but if a larger or smaller relative importance is required, we could change the definition of f_i. For now, however, we will continue with the logarithm.

If we compute for each signal for the package the ratio of $log\,(1 + S_i)$ and the maximum value $S_{i,max}$ achieved by all packages for S_i, we could develop a naive global criticality that orders the packages. We would have the scaling factor apply to the ratio, and after normalization have the formula:

$$C_{pkg} = \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i \frac{log(1 + S_i)}{log(1 + S_{i,max})}$$

with a criticality value in the range $[0, 1]$ as desired. This still has two problems however: it requires examining all signals for all packages to compute the criticality for just one package; and the value for one package can drift over time because of changes in other packages or in the set of signals or their weights.

We can address both of these issues with one more step. For a signal S_i we choose a threshold value T_i such that any signal value above that is at maximum importance. In other words, we clamp the signal values to a maximum value appropriate to the signal, considering anything above that to be truly "critical". For instance, we might consider any package with more than 100,000 dependents to be critical, and ignore the actual count for packages above that threshold.

The formula now becomes easier to compute, independent of other packages and, for truly critical packages, more stable:

$$C_{pkg} = \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i \frac{log(1 + S_i)}{log(1 + max(S_i, T_i))}$$

To summarize, for each signal S_i valid within a packaging system we choose two values, a scaling factor α_i representing the relative importance of that signal and a threshold value T_i representing the point at which that signal becomes of maximum importance. For each package, we compute a normalized sum of the ratio of the logarithm of that signal and the logarithm of the maximum of that value and the threshold.

This is the proposed criticality value for that package.

What if a signal is not available for a package? We have two options, and which to take may depend on the signal itself. We can either ignore that signal for that package completely, in which case the other signals that are

available rise in importance, or we can set it to zero, in which case its absence reduces the criticality of the package.