Dynamic Thresholds: Monitoring Network Reliability Online

Diane Lambert and Chuanhai Liu
Statistics Research
Bell Laboratories, Lucent Technologies
Murray Hill, N.J.
USA
dl@bell-labs.com, liu@bell-labs.com

Abstract

We describe a fast, statistically principled method for monitoring streams of counts that have long-term trends, rough evolving cyclical patterns, outliers and missing data. A key is to use a reference (predictive) model for the counts that captures their complex, salient features but has just a few parameters that can be kept up-to-date as the counts flow by, without requiring access to past data. This paper justifies using a negative binomial reference distribution with parameters that capture trends and patterns and that can be re-estimated quickly enough with each new count to keep up with the data flow. Changes in network reliability are detected by applying control chart methods to normal scores of the p-values of the counts under their reference distributions. Compared to thresholding isolated counts, control charting reduces the false alarm rate and increases the chance of detecting ongoing low level events. This *adaptive count thresholding* procedure performs well on both real and simulated counts.

1 Background

Communications networks may have thousands of network elements, such as routers and switches, that report statistics about their health on a regular schedule. Number of packets handled, number of packets dropped, and numbers of processing errors of various types may be reported every minute or second for every network element, for example. Such counts often become extreme when part of the network is in distress. Error counts may become too high, or number of packets carried may become too low. But what is the threshold for too high or too low? If the thresholds vary across network elements, then how can their information be combined to monitor network health? The first problem is often called monitoring and the second, which is considered harder, is a kind of data fusion.

Usually, network engineers choose thresholds by trial and error, hoping to balance the rate of missed events against the rate of false alarms. This is a delicate and daunting task because error counts and traffic volumes have time-of-day and day-of-week patterns, long-term trends, long tails, and geographic differences. Ignoring these complex patterns reduces the number of thresholds that have to be chosen, but at the expense of poor performance. On the other hand, treating each time period and region separately for each type of count leads to far too many thresholds to set by hand. As a result, thresholds that are set manually are often inappropriate, producing so many false alarms that only a small fraction can be investigated, seriously diluting the value of monitoring. There is a need for dynamic, adaptive thresholds that learn patterns in the counts and keep themselves up-to-date as data are collected.

Figure 1, for example, shows two weeks of five minute counts for one type of error for one network element. Counts of zeroes were numerous during this period, but counts higher than 40 were also frequent. Variability was high and increased as the median increased. Two smooths, computed using loess on the square root of the counts to tame variability and then squaring the fitted values to return to the original scale for plotting, are shown. The flatter, bolder curve corresponds to a large span ($f = 2/3$) and shows that the counts tended to increase even over this short two week period. The more detailed, thinner curve corresponds to a short span ($f = .025$) and shows that there was also a strong daily pattern, with a peak near 7 p.m. that rises more steeply than it falls. There is also evidence of a smaller peak later in the evening. About four weeks after the period shown here, there was a gap of 14 hours during which no data were collected. Clearly,

there is no one threshold that can be used to identify anomalous observations, regardless of time of day or even date. Nonetheless, the goal is to decide, with each incoming count, whether an event is in progress, without looking at all the past data, and without being distracted by the strong cyclical pattern, the high background variability, or the stretches of missing data.

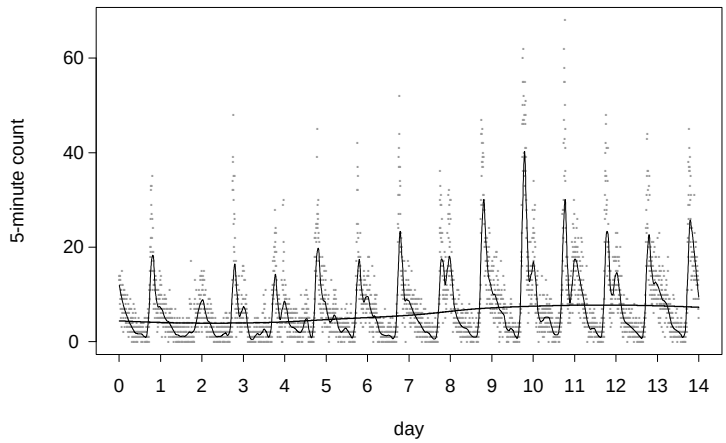

Figure 1: Two weeks of five minute error counts.

There are a few statistically inspired procedures for setting thresholds on network counts in the presence of rough cyclical patterns. One approach is to batch the data in intervals that are so short that the counts in an interval can be considered to be identically distributed. Thottan and Ji (1998), for example, fit an autoregressive process with constant mean and normal errors to the counts in an interval, and at the end of the interval compute a likelihood ratio to compare the fitted model for the current interval with the fitted model for the previous interval. If the ratio is large, then the current interval is flagged for a possible network event. The problem is that quick event detection requires short intervals, because detection occurs only at the end of an interval, but reliable event detection requires long intervals so the likelihood ratio test will be reliable.

Feather (1992) and Feather, Sieworek and Maxion (1993), in contrast, assume homogeneity not just within short time periods but also across cycles. Specifically, if a cycle is a day, then observations that occur in the same time period share a normal distribution with nearly the same parameters, regardless of the day. For example, a mean and variance may be stored for the [9:00, 9:15) a.m. period and updated by exponentially weighted moving averaging (EWMA) whenever a new count is obtained between [9:00, 9:15) a.m., even if the last count for that interval was taken yesterday. Thus, cycles are treated nonparametrically. But, learning is slow and thresholds are discontinuous because each time period is treated in isolation; a count at 8:55 a.m. has no effect on the estimated threshold for the interval from [9:00, 9:1).

2 Adaptive Count Thresholding

From a statistician's perspective, the first step towards devising a good monitor is to determine the shape of the distribution of counts when the network is in control. Extensive empirical model building, which is described in the full paper, shows that counts are Poisson distributed conditional on their means, and that variation in the mean is gamma distributed, after removing time-of-day patterns and trends. Mixing the Poisson with a gamma then leads to a negative binomial model. The negative binomial is parameterized by its moments to simplify online computations, and the moments are assumed to vary smoothly enough to be

interpolated from a coarse grid that captures the cyclical patterns in the data. The grid values are updated with each incoming count by EWMA to capture long-term trends in the data. The combination of quadratic interpolation between grid values and EWMA updated grid values smooths continuously over time, without arbitrary boundaries from one time interval to the next, and preserves cyclical patterns, without arbitrary assumptions about the shape of the pattern.

We do not advocate using quantiles of the reference distribution to threshold incoming counts, however. Even rare counts occur frequently if the data stream is fast enough, so thresholds on counts would need to be set at extreme quantiles to avoid excessive false alarms. But then low level but persistent degradation would be very difficult to detect. Thus, istead of formulating event detection as outlier detection, we formulate it as statistical process control and monitor a severity metric that retains information from previous counts. First, each count is approximately standardized by computing its tail probability under its negative binomial reference distribution. If the reference distribution were known exactly and if it were continuous, then this tail probability or p-value p_t would have a uniform[0,1] distribution for any reference distribution and its normal score $Z_t = \Phi^{-1}(p_t)$ would be normal$(0,1)$ under any reference distribution, regardless of the time-of-day or hour-of-week. The severity metric S_t is then an EWMA of the standardized counts $Z_t's$; S_t can be thresholded against a constant limit to detect periods of network degradation. (This gives a Q-chart in the terminology of statistical quality control.) Note that event magnitude and duration are both incorporated into the severity metric S_t. One extreme count (high magnitude, but short duration event) can push S_t beyond a threshold. Or, a sequence of many less extreme but still unusual counts (smaller magnitude, but long duration event) may push S_t beyond a threshold.

The raw p-values for the counts can also be used to monitor the thresholding system itself online. The p-values should be roughly uniform as long as a negative binomial reference distribution is valid and its estimated moments are appropriate. Thus, a histogram of the raw p-values should be roughly uniform if the reference distribution is appropriate. If it is not roughly uniform, then the monitoring process may need to be re-initialized. This kind of online validation is especially important in automated systems.

In short, adaptive count thresholding is reduced to four basic steps that are applied whenever there is an incoming count x_t. (1) *Interpolate* the stored grid values to obtain the estimated parameters for the reference negative binomial distribution F_t in effect at time t. (2) *Score* the count x_t by computing its p-value p_t under F_t and its normal score $Z_t = \Phi^{-1}(p_t)$; (3) *Threshold* the EWMA updated severity metric $S_t = (1-w)S_{t-1} + wZ_t$ (for some weight w in $(0,1)$ against a pre-specified constant threshold. (4) *Update* by EWMA the stored grid values with the count x_t or with a random draw from F_t if the count is an outlier or missing. Each step is quick to compute, so the processing needed for adaptive count thresholding can be completed before the next incoming count arrives.

3 An Example

Figure 2 shows the results of thresholding two weeks of five minute counts after a network element had been monitored for three weeks. (Behavior during initialization and later weeks and results for simulated count data are discussed in the full paper.) Each panel in Figure 2 is labeled by a date and divided into an upper and lower region. The upper region shows the counts (on a square root scale to make better use of the plotting region, although the thresholding procedure is applied to the raw counts) and the evolution of the .0001, .5 and .9999 quantiles of the reference distribution as data are collected. The .0001 quantile is often zero. As would be expected, the median generally runs through the bulk of the counts. The lower region of each panel shows the severity statistic S_t (in red) evolving as new counts are obtained. The thresholds for setting an alarm are shown as horizontal black lines in the bottom region. When S_t exceeds the threshold, the corresponding count x_t is colored red and the background on the strip label for the day is darkened.

In Figure 2, the severity S_t floats near zero, as it should when there is no event, even though there are isolated counts beyond the .9999 quantile on January 13, 14 and 17. These counts would generate false alarms if individual counts were thresholded at the .9999 quantile. On January 23, S_t exceeds the threshold for a few hours, and many counts are far beyond the medians of their reference distributions. However, this discrepancy is subtle in an absolute sense and would not be apparent without a good description of the

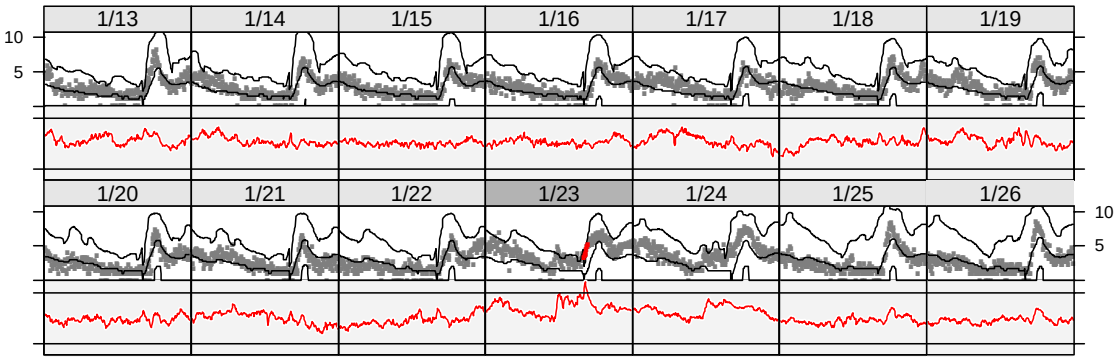

Figure 2: Two weeks of monitoring network error counts. The red line shows the evolution of the severity metric S_t. The red dots show counts (on the square root scale) that correspond to value of the severity metric that exceed the threshold.

daily pattern. Also note the effect of the outliers on the updated reference distributions. The outliers have little effect on the medians for the next day, but the corresponding .9999 quantiles are increased because uncertainty about the behavior of the counts has increased. These effects are desirable: outliers should not be able to move the estimated median of the reference distribution quickly (even though it is computed from mean and variance estimates), but outliers should stretch the tail of the reference distribution. The effect of outliers is dampened because any outlier is replaced with a random draw from the tail of the reference distribution. Here, any point beyond the .9999 quantile was marked as an outlier and replaced with a random draw from beyond the .99 quantile of the reference distribution. Skipping the updating step when there is an outlier, instead of replacing it by a random draw from the tail of the reference distribution, would underestimate the tails of the reference distribution, which would produce too many false alarms.

4 Discussion

Network reliability is complex, and monitoring a single series of counts is only one step towards detecting network degradation. Typically, many series of counts are monitored for each network element, and there are multiple elements to monitor. Because adaptive count thresholding provides a way to standardize counts, taking into account differences in the reference distribution caused by cyclical patterns and long-term trends, it takes a step towards monitoring multiple streams of counts, whether for one or many network elements. Namely, each sequence of counts can be reduced to a sequence of severity scores S_t which roughly follow a gaussian autoregressive process with mean zero. Combining the scores then requires monitoring their correlations. This approach to combining information across network elements assumes that each network element continually passes its S_t stream to a central site. Transferring this much data is unrealistic in many applications though, especially those involving low power sensors. Instead, it is more likely that S_t or some summary of S_t would be transferred only when S_t is "interesting". Appropriate ways to combine streams in the presence of that kind of censoring are a topic of future research.

References

Feather, F. W. (1992). *Fault Detection in Ethernet Networks via Anomaly Detection*, Ph.D. thesis, Department of Electrical Engineering and Computer Engineering, Carnegie Mellon University.

Feather, F. W., Siewiorkek, D. and Maxion, R. (1993). Fault detection in ethernet network using anomaly signature matching, *Proceedings of the SIGCOMM93*, 279–288.

Ji, C. and Thottan, M. (1998). Proactive anomaly detection using distributed agents, *IEEE Network*, 21–27.