Anomaly Labeling Instructions

The objective is to indicate the timestamps at which an anomaly is identified to start, according to the instructions below. Labels should not be done for data files in the directory nab/data/realKnownCause.

1. The first 15% of a data set is the *probationary period*. Do not label any anomalies inside the probationary period, but do observe the data patterns inside the probationary period – these patterns are learned as *normal* so if you see these patterns again in the rest of the data, they are not anomalies; that is unless a new *learned normal* occurs. In the data visualizer, the probationary period has been shaded grey.
2. If the normal pattern changes in the data (after the first 15%), the first data point of the change is the start of an anomaly – e.g. point "a" in Fig. 1 below. A new pattern in the data is established as the *new normal* if it proceeds for approximately 10% of the data file length – e.g. at point "b" in Fig. 1 below, the new pattern, a straight line, has been learned.
3. If after a new normal is established, the data stream goes back to a previously learned pattern – i.e. a *remembered normal* – then this transition is not an anomaly. E.g. in Fig. 1 below, the original data pattern resumes at point "c"; this is a remembered normal and should not be labeled as an anomaly.
4. A long flat line becomes an anomaly if the period of flatness exceeds the longest period that was in the normal data before that point. For instance, in Fig. 2 below, some periods of the data at zero are learned in the probationary period; the green arrow on the left shows the longest zero-period that is learned. When another, longer zero-period occurs after the probationary period is over (the green arrow on the right), the anomaly should be labeled starting after the length of the previous longest learned period, at point "a".
5. A significantly different data value – i.e. more than 20% different – can be labeled an anomaly, even if the value isn't a new maximum or minimum value. For instance, in a normal pattern of spikes that reach values of 10 and 60, a new spike that reaches 30 is an anomaly, even though this spike's value is between the normal peaks.
6. Don't over analyze trying to locate anomalies. In general there should be very few anomalies per data file. One test to use: eliminate all future data from your thinking and ask yourself, "Assuming this is an important streaming data source, would you wake someone up if this occurred?"
7. If there is a lot of noise in the data, a pattern change has to be more obvious to be an anomaly. In very clean data, almost any new value would be anomalous; in noisy data it isn't.
8. Ignore all "future" data as you move through a dataset looking for anomalies. That is, try to identify the start of an anomaly in real-time, as a detector would.

9. If in doubt whether there are one or two anomalies in a relatively small window, merge two anomalies into one.
10. Save the labels file as a JSON with naming convention 'XY_labels_vAB.json', where 'XY' are your initials and 'vA.B' refers to the corresponding NAB version number. The format should be a dictionary of key-value pairs for each data file; the key is the data file location, and the value is a list of timestamps indicating the anomaly start times. The script nab/scripts/create_empty_label_file.py will create an empty JSON file for labels, and for an example please see nab/labels/raw/AL_labels_v1.0.json.
11. Keep in mind that anomalies are rare by definition, and this is an underlying assumption of NAB parameters and logic.
12. Anomalous data points within close proximity are assumed to represent the same anomalous data, so please choose one of the data points as your label. Here we define close proximity as 1% the length of a data file. If your raw labels contain timestamps that are within close proximity, an error will be raised that requests you fix the labels.

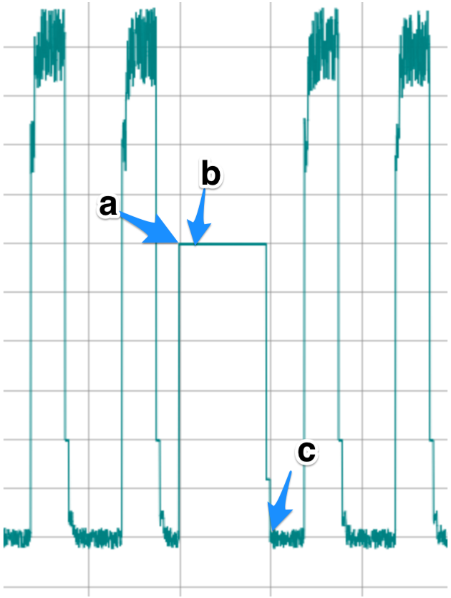

Fig. 1

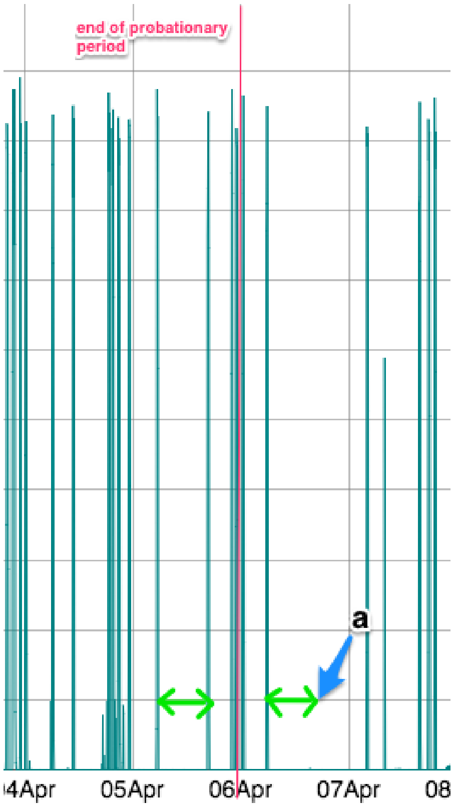

Fig. 2