GENERALIZED BOUNDARY DETECTION USING COMPRESSION-BASED ANALYTICS

Christina Ting, Richard Field, Jr., Tu-Thach Quach, and Travis Bauer

Sandia National Laboratories
Albuquerque, NM 87123

ABSTRACT

We present a new method for boundary detection within sequential data using compression-based analytics. Our approach is to approximate the information distance between two adjacent sliding windows within the sequence. Large values in the distance metric are indicative of boundary locations. A new algorithm is developed, referred to as sliding information distance (SLID), that provides a fast, accurate, and robust approximation to the normalized information distance. A modified smoothed z-score algorithm is used to locate peaks in the distance metric, indicating boundary locations. A variety of data sources are considered, including text and audio, to demonstrate the efficacy of our approach.

Index Terms— Signal processing, Machine learning, Change detection, Information distance, Peak finding

1. INTRODUCTION

Let z be a sequence of tokens made available over time. This general class of data is prevalent in, for example, computer network traffic, text, and audio signals. Identifying structure in z requires significant latency and domain knowledge of the underlying data. Our objective herein is to develop a general, unsupervised approach for determining locations within z where the information content suddenly and substantially changes; we refer to these locations as structural boundaries.

The normalized information distance (NID) is the optimal universal distance metric to capture the differences between two sequences [1]. It is, however, non-computable. We seek a method that provides an accurate approximation to the NID but is also computationally efficient; the latter is necessary due to the nature of our sliding application, where changes in information content need to be approximated repeatedly as we progress through z. Further, we require our method to be robust to small changes in information content; an algorithm that violates this requirement will produce a noisy signal that renders boundary detection difficult.

We propose a new algorithm, referred to as *sliding information distance* (SLID), that provides a fast, accurate, and robust approximation to the NID. When combined with a smoothed z-score algorithm for peak finding, this approach provides a new method for boundary detection within sequential data. Although other approaches may perform better for specialized applications where domain knowledge is available, our approach is general and applicable to a wide variety of data sources. Further, it is possible to extend these concepts to streaming applications or to datasets in higher dimensions, e.g., edge detection in images and flaw detection in engineering materials.

2. PREVIOUS WORK

We provide a brief overview of existing boundary detection techniques on time-series data. Our focus is on unsupervised approaches that do not require training data. General boundary or change-point detection often involves sliding adjacent windows over time series data, collecting statistics of the underlying data within each window, and computing a distance function that operates on the statistics to determine large distances between adjacent windows [2, 3, 4]. For domain-specific data, such as audio, specialized methods can also be used.

To the best of our knowledge, a specialized information-theoretic distance metric for boundary or change-point detection does not exist. However, there has been extensive work on approximating the NID, most of which are based on the normalized compression distance (NCD) [6]. NCD uses standard compression algorithms and is therefore easy to implement in practice, but is too costly for a sliding application. It has been shown that NCD can be approximated by operations on the underlying dictionaries, thereby bypassing the compression step and improving the computation speed [7, 8, 9, 10, 11]. SLID is a variant of these methods, where the dictionary construction has been further optimized for a sliding boundary detection application. Although we could have chosen to modify any of these methods, SLID builds off the Lempel Ziv Jaccard Distance (LZJD) [11] due to the simplicity of the Jaccard index as the distance metric.

3. METHODS

3.1. Sliding information distance (SLID)

Let k denote a position within the sequence z. We formulate the boundary detection problem by considering two adjacent

subsequences of z, denoted by x_k and y_k, each of length $w \geq 1$:

$$z_0 \cdots z_{k-w-1} \underbrace{z_{k-w} \cdots z_{k-1}}_{x_k} \underbrace{z_k \cdots z_{k+w-1}}_{y_k} z_{k+w} \cdots$$

We denote the SLID score of z at position $k = w, \ldots$ by

$$S_k(z; w) = 1 - \frac{|D(x_k) \cap D(y_k)|}{|D(x_k) \cup D(y_k)|}, \tag{1}$$

where $D(x)$ denotes a set representation of the Lempel-Ziv (LZ) [12] dictionary encoding of sequence x. The right hand side of Eq. (1) is the Jaccard distance between two LZ sets, and S_k takes values in $[0, 1]$.

Algorithm 1 Sliding information distance.

function SLID(sequence z, window size w)
  $S \leftarrow [0]$ ▷ list initialized with $w-1$ zeros.
  for $k = w, \ldots$ **do**
    $x_k \leftarrow z_{k-w}, \ldots, z_{k-1}$
    $y_k \leftarrow z_k, \ldots, z_{k+w-1}$
    if $k == w$ **then**
      Lx, $Dx \leftarrow$ makeLZdict(x_k)
      Ly, $Dy \leftarrow$ makeLZdict(y_k)
    else
      Lx, $Dx \leftarrow$ updateLZdict($x_k[-1]$, Lx)
      Ly, $Dy \leftarrow$ updateLZdict($y_k[-1]$, Ly)
    end if
    S.append($1 - |Dx \cap Dy|/|Dx \cup Dy|$)
  end for
  return S
end function

Algorithm 2 Initialize the LZ dictionary.

function MAKELZDICT(sequence b)
  $\ell \leftarrow [\,]$, start $\leftarrow 0$, end $\leftarrow 0$
  while end $<$ $|b|$ **do**
    item $\leftarrow b$[start : end]
    if item $\notin \ell$ **then**
      start $\leftarrow$ end
    end if
    ℓ.append(item)
    end $\leftarrow$ end + 1
  end while
  return ℓ, set(ℓ)
end function

Algorithm 1 presents the main computation of SLID. At initial position $k = w$, the first step is to apply Algorithm 2 to compute the ordered LZ list of subsequences for *each* token in x_w and y_w, followed by the Jaccard distance of the set representation of the LZ list. We maintain the state of the ordered LZ list, and not simply the set, so that when the position is progressed to $k = w + 1$, we can update the LZ list according to Algorithm 3. Here, instead of recomputing the LZ sets of x_{w+1} and y_{w+1}, we drop the first entry in the LZ list and append a new entry for the last token in the new subsequence. Once again, S_{w+1} operates on the set representation of the updated LZ lists.

Algorithm 3 Update the LZ dictionary.

function UPDATELZDICT(token t, list ℓ)
  if $\ell[-1] \notin \ell[0 : -2]$ **then**
    item $\leftarrow t$
  else
    item $\leftarrow \ell[-1] + t$
  end if
  $\ell \leftarrow \ell[1 :]$
  ℓ.append(item)
  return ℓ, set(ℓ)
end function

Applying Algorithm 3 to update the LZ list has two benefits. First, the computational cost of Algorithm 2 is at least $O(w)$ as each token in the subsequence of length w has to be parsed. This cost is $O(1)$ for Algorithm 3. Second, the updating step ensures smaller changes in the LZ set over recomputing them; this results in a smoother SLID score for boundary detection.

3.2. Boundary detection

As SLID is being computed for a given sequence, the boundary locations can be estimated by identifying anomalous peak regions in SLID. We apply the smoothed z-score algorithm [13], an unsupervised change detection algorithm, to locate these peak regions. Briefly, the algorithm identifies a point as anomalous if it exceeds a threshold of n standard deviations over a running mean of the previous m data points; we choose $n = 2$ and $m = 64$. Anomalous points are assigned an influence of 0.001 (see [13] for details on the influence). We define K, a set of contiguous anomalous positions, as a peak region for which a boundary exists.

We execute the smoothed z-score algorithm in the sequence, in the directions of increasing (forward) and decreasing (reverse) k to produce a collection of contiguous anomalous positions, denoted by sets K_f and K_r, respectively. We define the peak region by $K = K_f \cap K_r$. Assuming K is a nonempty set of contiguous positions,

$$k^* = \arg\max_{k \in K} S_k(z; w) \tag{2}$$

is our estimate for the boundary location k^* in K. For performance assessment, we assume that boundary estimates that are within 16 tokens of the true boundary are true positives. Further, we require that $|K| > 16$ to reduce false positives. In practice, this algorithm will produce a collection of K, each containing a boundary.

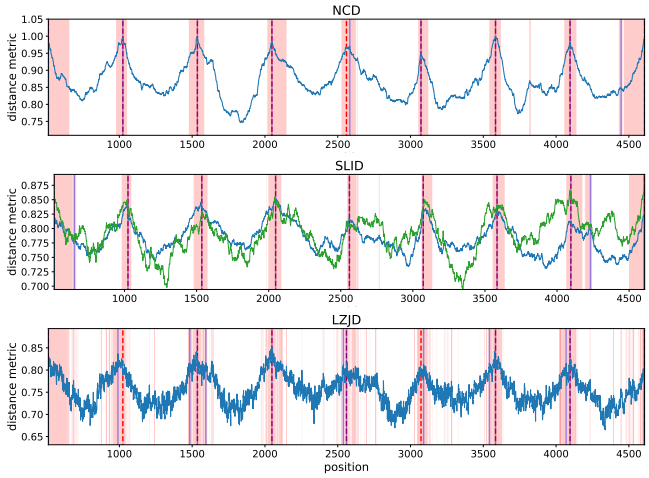

Fig. 1. Boundary detection using NCD, SLID, and LZJD on written text with a sliding window of width $w = 512$ (blue) and the peak-finding algorithm. SLID using $w = 256$ (green) is also shown for comparison.

Table 1. Time, noisiness (σ), and correlation with NCD for the results presented in Fig. 1.

	Time (seconds)	σ	NCD Correlation
NCD	1100	0.0019	1.00
SLID	0.2	0.0016	0.78
LZJD	3.6	0.0123	0.77

4. DATA

To validate our algorithm against multiple data sets, we synthesize sequences with known ground truth for boundaries using multiple data sources. Our first data set contains sections of text randomly selected from English and Spanish translations of a United Nations (UN) document [14]. A second data set contains sections of audio randomly selected from a male and a female speaker from the LibriSpeech ASR corpus [15, 16]. Each data set contains 100 ground truth boundaries with section lengths $s = 512$ tokens for quantifying algorithmic performance and computing precision-recall curves.

5. RESULTS

We begin by presenting results on identifying language boundaries in UN documents using NCD, SLID, and LZJD, another dictionary-based distance metric not optimized for a sliding boundary detection [11]. Although we use PPMAC [17] as the compressor for NCD, we note that we have tried alternative fast compression algorithms and observed a decrease in the performance of NCD.

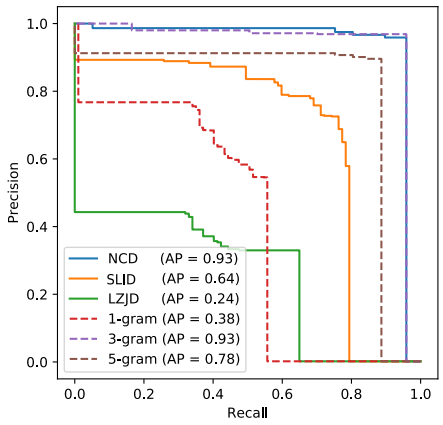

Fig. 2. Precision-recall (P-R) curves for the UN document data set.

Figure 1 shows boundary detection results for each method using a window width $w = 512$. The positions of true boundaries are denoted by the vertical dashed red lines. As mentioned in Section 4, every multiple of $s = 512$ is a true boundary. The shaded red regions indicate the sets K defined by Eq. (2) for which the smoothed z-score algorithm indicates anomalous positions, and the vertical blue lines indicate k^*, the boundaries identified within K. Thus any instance of a blue line corresponding with a red dashed line is a true positive; any instance of a blue line not corresponding to a red-dashed line is a false positive.

By visual inspection of Fig. 1, all compression-based distance metrics provide some indication of boundaries, as indicated by peaks in the score near (or at) a true boundary. Although we have selected a window size w equal to the section length s, we also show SLID results using a window size $w = 256$ (green) to demonstrate that our method successfully produces peaks over a range of window sizes. Future work will involve extending SLID to automatically determine the optimal window size.

Table 1 summarizes the performance of each method with respect to the three qualities we desire in a boundary detection scheme: (1) efficiency, (2) smoothness, or robustness to small changes in information content, and (3) the ability to accurately approximate the NID. Both LZJD and SLID run three and four orders of magnitude, respectively, faster than NCD, and are still accurate approximations for the NID, as quantified by the correlation coefficient with the NCD. Further, let $\delta_k = d(k) - d(k-1)$ for $k = w, \ldots$, where we use $d(k)$ to denote a general distance metric at location k in the sequence. SLID yields a much smaller standard deviation, σ, in δ_k compared to LZJD, indicating a smoother signal. As shown qualitatively in Fig. 1, a noisy LZJD (lower panel) produces several false positives, whereas SLID (middle panel) produces a notably smoother score for improved boundary detection.

To assess performance, Fig. 2 presents a precision recall

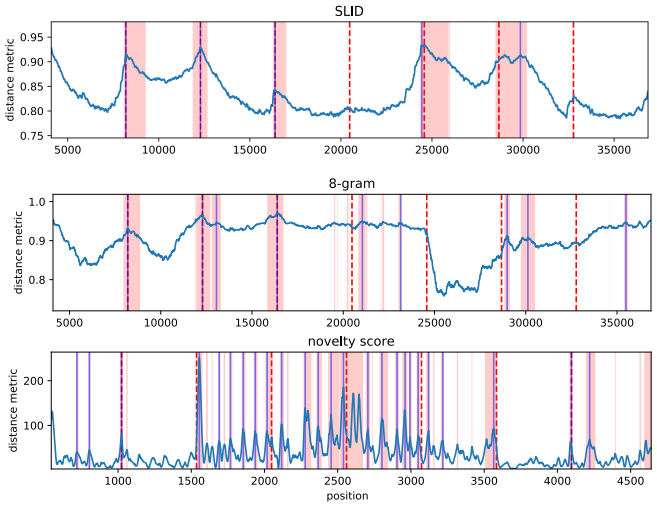

Fig. 3. Boundary detection using SLID, 8-gram, and novelty score on audio files with a sliding window of width $w = 512(\times 8)$.

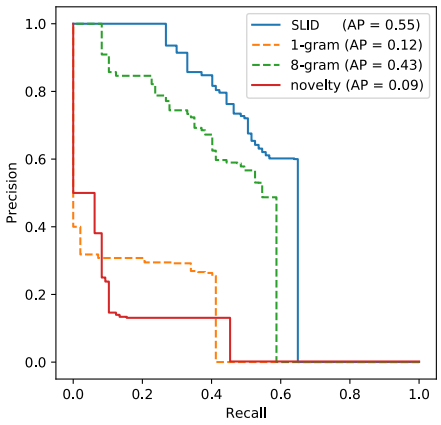

Fig. 4. Precision-recall (P-R) curves for the audio data set.

(PR) graph with average precision (AP) scores. These results are obtained from the full dataset constructed from 100 segments of length $s = 512$ chosen from the two translations of the UN document. NCD outperforms all other metrics but is extremely slow; SLID significantly outperforms LZJD (compare AP $= 0.44$ with AP $= 0.21$). As discussed above, this result can primarily be attributed to the smoother signal for SLID and therefore far fewer false positives.

For comparison to more traditional text-based methods, Fig. 2 also presents results for the cosine distance over adjacent sliding windows of n-gram distributions. The 3-gram distance, which is commonly used to describe written language [18], performs as well as NCD. In contrast, the 1-gram distance cannot capture the full complexities of the structural differences between Spanish and English text, and the 5-gram distance cannot capture the statistical distribution within the sliding window of width $w = 512$. If the underlying data can be well-described by an n-gram, as is the case with text, then it is not surprising that an n-gram approach (with the appropriate n) can outperform a general compression-based method. However, in general, some knowledge of the underlying data is needed to select the optimal n-gram; compression-based methods are a good choice where there is no such knowledge.

Finally, we apply SLID to an audio dataset for which we do not expect an n-gram to be the appropriate description of the underlying data. In Fig. 3, we present results for SLID, together with an n-gram approach and a specialized method developed to detect novelty in audio data [5]. Red dashed lines correspond to true boundaries and blue lines correspond to estimates k^* as defined in Eq. 2. Because each floating point value of the audio file is represented by eight bytes, the horizontal axes of SLID and the 8-gram are eight times the scale of the novelty score, which is computed using floating point values. By Fig. 3, we observe that the peaks produced by SLID are more pronounced at the true boundaries than the peaks produced by the 8-gram approach. Furthermore, there are fewer false positives, particularly when compared with the novelty score, where the latter seems to measure local changes in a given frequency. As a result, the PR curve in Fig. 4 shows that SLID outperforms both the n-gram and the novelty methods.

6. CONCLUSIONS

We have presented a fast, efficient, and robust compression-based method for detecting boundaries in arbitrary sequences of data, including data streams. We have demonstrated the versatility of our approach through several experiments on multiple data sources. As our method is computationally efficient, it is possible to apply different window sizes to obtain a multi-scale structural representation of the underlying data source for boundary detection. This could lead to further improved detection performance or provide the ability to identify coarse and fine-grained boundaries. We defer investigating this problem to our future work.

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2019-1500 C.

7. REFERENCES

[1] P. M. B. Vitányi, F. J. Balbach, R. L. Cilibrasi, and M. Li, "Normalized information distance," in *Information Theory and Statistical Learning*, pp. 45–82. Springer, 2009.

[2] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems*, vol. 51, pp. 339–367, 2017.

[3] A. Aue, S. Hörmann, L. Horváth, and M. Reimherr, "Break detection in the covariance structure of multivariate time series models," *The Annals of Statistics*, pp. 4046–4087, 2009.

[4] Y. Kawahara and M. Sugiyama, "Sequential change-point detection based on direct density-ratio estimation," *Journal Statistical Analysis and Data Mining*, pp. 114–127, 2012.

[5] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *IEEE International Conference on Multimedia & Expo (ICME)*, 2011.

[6] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi, "The similarity metric," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3250–3264, 2004.

[7] A. Macedonas, D. Besiris, G. Economou, and S. Fotopoulos, "Dictionary based color image retrieval," *Journal of Visual Communication and Image Representation*, vol. 19, no. 7, pp. 464–470, 2008.

[8] D. Cerra and M. Datcu, "A fast compression-based similarity measure with applications to content-based image retrieval," *Journal of Visual Communication and Image Representation*, vol. 23, no. 2, pp. 293–302, 2012.

[9] A. Bogomolov, B. Lepri, and F. Pianesi, "Generalized Compression Dictionary Distance as Universal Similarity Measure," *ArXiv e-prints*, Oct. 2014.

[10] H. Koga, Y. Nakajima, and T. Toda, "Effective construction of compression-based feature space," in *2016 International Symposium on Information Theory and Its Applications (ISITA)*, Oct 2016, pp. 116–120.

[11] E. Raff and C. Nicholas, "An alternative to NCD for large sequences, Lempel-Ziv Jaccard distance," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[12] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[13] J.-P. van Brakel, "Smoothed z-score algorithm," http://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data, 2016, Accessed: 2018-09-19.

[14] "Children and armed conflict: Report of the secretary-general," http://research.un.org/en/docs/find/reports, 2018, Accessed: 2018-07-26.

[15] V. Panayotov and D. Povey, "Open speech and language resources," http://www.openslr.org/12/, 2016, Accessed: 2018-07-26.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.

[17] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.

[18] Peter F. Brown, Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.