Ranked Leakage Sensitive Hashing

Rohith Prakash

1 Introduction and Definitions

Consider streams of time series describing the utilization traces of a set of n resources, each at some (fixed) granularity. Intuitively, time series which are “closer” to each other (by some metric d) will be harder to distinguish by an arbitrary classifier and present fewer possibilities for information leakage. We expand on this intuition by considering a hashing family $\mathcal{H}$ and hashing function $h \in \mathcal{H}$ which has the following properties:

(i) The hashing function *collides* ($h(x) = h(y)$) with high probability iff x and y are “close” (i.e., $d(x, y) < r$ for some fixed r)

(ii) The hashing function has few *false collisions*: If $d(x, y) > r$, $\Pr[h(x) = h(y)] < p$ for some fixed, small p.

(iii) The hashing family is parametrizable by distance r. A family $\mathcal{H}_r$ obeys (i) and (ii) with collision distance r.

Properties (i) and (ii) corroborate the notion of closeness to collision probabilities, which allows for the grouping of similar time series. Property (iii) allows for an iterative hashing scheme to *rank* the probabilities of closeness based on the varied parameter r. This is useful for classification (modeling an intelligent adversary)

1.1 Distance Metric

Denote the space of a time series as S, and define a function d:

$$\begin{aligned} d_{a,b} \colon S^2 &\to \mathbb{R}_+ \\ (x[a:b], y[a:b]) &\mapsto r \end{aligned} \tag{1}$$

Where d maps a portion of two time series to a non-negative real number that represents the distance between the partial time series.

Additionally, we require d to actually be a metric [1]:

$$\begin{aligned} d(x, y) = 0 &\Leftrightarrow x = y \\ d(x, y) = d(y, x) &\geq 0 \\ d(x, y) \leq d(x, z) &+ d(z, y) \end{aligned} \tag{2}$$

Simple Euclidean distance will not suffice, as it has many limitations for time series comparison [2]. Dynamic Time Warping (DTW) presents a much more accurate method for comparing time series, but it does not satisfy the triangle inequality and is thus not a metric. We could leverage an approximate lower-bound DTW [3] as this satisfies the triangle inequality, but recent literature [4, 5, 6] provide additional metrics which have proven efficiency and accuracy in fields such as image processing and classification. Reproducing kernel Hilbert spaces have been studied with respect to locality sensitive hashing methods and may be of great use to this problem.

1.2 Reproducing Kernel Hilbert Space

Reproducing kernel Hilbert spaces (RKHS) are a Hilbert spaces (vector spaces over which inner products are defined) defined over functions with the following property:

(i) For any two functions $f, g \in H$ defined over $dom(f)$, $\|f - g\|_H < \epsilon \Rightarrow |f(x) - g(x)| < \delta \; \forall \; x \in dom(f)$.

We consider time series to be discrete-time functions in an arbitrary RKHS and leverage kernelized LSH schemes [7, 5, 4] to perform our ranked hashing. Since RKHS have a useful norm which intrinsically arises from their construction, we can leverage the norm in constructing a kernelized hash family to perform an explicit embedding. Note, however, that this norm is not necessarily a Euclidean, point-wise norm. The reverse, $|f(x) - g(x)| < \delta \Rightarrow \|f - g\|_H < \epsilon$, need not be true in this space.

2 Hashing model for privacy

Consider a database of n samples $D = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ with feature map:

$$\Phi \colon \mathbb{R}^d \to \mathcal{H} \tag{3}$$

where $\mathcal{H}$ is a RKHS. A kernelized hashing scheme for a similarity metric requires that each hash function h_r satisfy:

$$\Pr[h_r(x_i) = h_r(x_j)] = \kappa_r(x_i, x_j) \tag{4}$$

where $\kappa(\cdot, \cdot)$ is a kernel function defined by:

$$\kappa(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \tag{5}$$

We assume κ_r is normalized with respect to distance bias r such that $\kappa_r(\cdot, \cdot) \in [0, 1]$, and κ_r follows properties (i), (ii), and (iii) of Section 1.

Note that equation 4 confers similar "bounds" to the differential privacy condition [8]:

$$\Pr[f(x) = o] \leq e^{\epsilon} \Pr[f(y) = o] \tag{6}$$

Intuitively, this bounds the probability ratio of an observed output o being the result of inputs x or y to some function f. While the hashing metric in equation 4 does not give a hard bound, it instead allows the evaluation of privacy leakage and the clear separation of time series into ranked buckets by varying distance parameter r.

Consider a hash family $\mathcal{H}$ which approximates a metric for time series distance and is parametrizable by distance threshold r for collision probability (see properties (i), (ii), and (iii) in Section 1).

Let $\mathcal{R} = \{r_1, r_2, \ldots\}$ be a set of stratifying distance thresholds, and let S be the space of all possible time series. Define an equivalence relation $\sim_r$ on S where $x \sim_r y \Leftrightarrow \kappa(x, y) < r$. Then, π_r is a projection mapping of $\sim_r$ such that $\pi_r \colon X \to X/\sim_r$, where $\pi_r(x) = \pi_r(y)$ iff $\kappa(x, y) < r$.

Iteratively applying LSH with hash functions $h_r \in \mathcal{H}_r\ \forall r \in \mathcal{R}$ will approximate the projections π_r for each threshold. Let $L_r = \{[h_r(x)], \ldots\}$ be the set of LSH buckets (the set unique hash mappings of h_r). If $h_r \approx \pi_r$, then L_r will closely approximate the quotient $X/\sim_r$ and thus $\{L_r | r \in \mathcal{R}\} \approx \{X/\sim_r\}$.

2.1 Anomaly Detection

What this theory describes is that similar time series can be grouped together and ranked by their closeness by kernel function κ [9]. In the context of anomaly detection, we can apply this scheme to a set of *normal* traces (time series of utilization, label encoded syscalls, etc.) to determine what thresholds and buckets constitute normal execution with a finer granularity. Instead of a binary label of *normal* vs *anomalous*, we can instead stratify applications by types of execution. We speculate that this makes our detection algorithm more robust to adversarial interference.

Given the initial LSH bucketing of normal traces, we can continuously apply the hashing scheme to testing traces over a rolling window to categorize their behavior. Traces which do not match enough previously "normal" buckets over the set of thresholds may be considered to represent anomalous activity.

2.2 Unsupervised, Intelligent Adversary

A similar approach to Section 2.1 can be applied without a predetermined training set for time series classification through clustering. Adjusting the set of threshold values confers the notion of closeness, which may be used to determine which sets of output traces resulted from the same input to a program, for example.

3 Proposed Metrics

Consider the Minkowski distance:

$$M(x, y) = \Big(\sum_k |x_k - y_k|^p\Big)^{1/p}$$

For $p \geq 1$, $M(\cdot, \cdot)$ is a metric due to Minkowski's inequality [10, p. 190]. Additionally, for $p = 2$, this is the standard Euclidean distance. Like the Euclidean distance, this distance metric fails to account for phase and shape changes between time series.

To this end, we consider two different metrics based on a p-dimensional Minkowski distance. The first is due to Batista et al. [11] which attempts to account for time series complexity and has been shown to greatly improve the mean accuracy rates of time series comparison [12]. Consider the following *complexity measure*:

$$C(x) = \sqrt{\sum_i (q_i - q_{i+1})^2} \tag{7}$$

Using this complexity measure, we define a *complexity-invariant* distance metric based on the Minkowski distance $M(\cdot, \cdot)$:

$$d(x, y) = M(x, y) * \frac{\max(C(x), C(y))}{\min(C(x), C(y))} \tag{8}$$

The second metric we consider attempts to correlate Minkowski distance across time with an additional penalty term for time series which are out of phase:

$$d(x, y) = \min_{i,j<\frac{n}{2}} \{M(x[i:N], y[j:N]) + c * (i + j)\} \tag{9}$$

where x, y have length n and $N = n - \max(i, j)$. c is a scalar penalty factor for out of phase alignment between x and y.

For both of these metrics, we consider p-dimensional Minkowski distance where the original time series have length p.

3.1 Proposed Hash

Let S be the space of n-point time series to be processed. Pick a random lines in $\mathbb{R}^2$ and project each point x_i onto each line h_k. Each line will be partitioned into buckets of size $\frac{r}{n}$ (here, we see the parametrization of the hash). In practice, we fix a number M (say, 2^{32}), and apply $h_r(x) = \frac{\pi(x)*n}{r} \bmod M$.

Points are considered to intersect if they hash to the same bucket on a fixed number of hash function. We propose that the number of hash functions on which points must collide to determine intersection be a tunable parameter. Time series which intersect on more than $\frac{n}{2}$ points (in time order) are said to be candidate pairs for r-closeness.

4 Key Takeaways

- Use hash that approximates time series distance.
- Compose multiple kernels or hash families to obtain ranked "normality" metric.
- Apply ranking to classification by an unsupervised classifier to determine amount of leakage possible without prior knowledge.
- Apply ranking to normality (anomaly) detection.
- Perform continuous hashing on rolling windows of execution across channels of multiple resources to classify activity in real-time.
- To be considered: Overlay results with ShapeGD.

References

[1] M. Rosenlicht. *Introduction to Analysis.* Dover books on mathematics. Dover Publications, 1968.

[2] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recogn.*, 44(9):2231–2240, September 2011.

[3] Daniel Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recogn.*, 42(9):2169–2180, September 2009.

[4] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1092–1104, June 2012.

[5] Ke Jiang, Q. Que, and B. Kulis. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4933–4941, June 2015.

[6] Y. B. Kim, E. Hemberg, and U. M. O'Reilly. Stratified locality-sensitive hashing for accelerated physiological time series retrieval. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2479–2483, Aug 2016.

[7] D. C. Kale, D. Gong, Z. Che, Y. Liu, G. Medioni, R. Wetzel, and P. Ross. An examination of multivariate time series hashing with applications to health care. In *2014 IEEE International Conference on Data Mining*, pages 260–269, Dec 2014.

[8] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[9] H. Hachiya and M. Matsugu. Nsh: Normality sensitive hashing for anomaly detection. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 795–802, Dec 2013.

[10] R.L. Wheeden. *Measure and Integral: An Introduction to Real Analysis, Second Edition.* Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis, 2015.

[11] Gustavo E. Batista, Eamonn J. Keogh, Oben Moses Tataw, and Vinícius M. Souza. Cid: An efficient complexity-invariant distance for time series. *Data Min. Knowl. Discov.*, 28(3):634–669, May 2014.

[12] Rafael Giusti and Gustavo E. A. P. A. Batista. An empirical comparison of dissimilarity measures for time series classification. In *Proceedings of the 2013 Brazilian Conference on Intelligent Systems*, BRACIS '13, pages 82–88, Washington, DC, USA, 2013. IEEE Computer Society.