Stratified Locality-Sensitive Hashing for Sublinear Time Critical Event Prediction

Y. Bryce Kim Erik Hemberg Una-May O'Reilly
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
32 Vassar St, Cambridge, MA 02139
{ybkim,hembergerik,unamay}@csail.mit.edu

1 Motivation

In data-driven medicine, fast yet accurate prediction of acute and critical events based on time series signal data from patient monitors is crucial especially in intensive care units (ICU). In such setting, promptness is demanded, so if a task can be completed dramatically faster, it is often acceptable to tolerate a modest amount of approximation. We are particularly interested in the problem of making a prediction on the future state of a query based on the result of a quick retrieval of waveforms similar to the query. In [1,2], we introduced a scalable prediction system based on locality-sensitive hashing (LSH) [3] for high-dimensional massive physiological data. The prediction based on LSH is essentially a two-step process of first quickly retrieving "patients with trajectories like mine", the nearest neighbors (NNs) of our query of interest by LSH, and second, extrapolating the information of NNs for prediction via the majority vote. LSH is a sublinear time, *approximate* search method enabling a quick retrieval of a small approximate NN set. It is based on the idea of using a specialized hashing method to provide preliminary filtering of NN candidates to reduce the time cost of a follow-up linear search among them. LSH intrinsically introduces the trade-off between the level of approximation and speed. In comparison to the *exact*, linear k-nearest neighbor (KNN) search, our system based on approximation via LSH offers a significantly faster querying time even though its drop in prediction accuracy is trivial. Approximation of NNs is justifiable because even in exact search, a distance measure used is also only an approximation to the ground truth.

When utilizing LSH, the appropriate choice of distance function for measuring similarity is critical because of the one-to-one relationship between the distance function and its unique corresponding family of locality-sensitive hash functions. Typically, one locality-sensitive hash family offers only a single perspective on the data (for example, amplitude, shape, or angle) with its associated distance function. However, interpreting clinical physiological waveforms requires diverse perspectives on the data. Both the amplitude (e.g. blood pressure level) and the shape of waveforms (e.g. trend and cycle frequency) contain important clinical information. For example, acute hypotensive episode (AHE) is defined as a sudden dropping (shape) of blood pressure to below 60 *mmHg* (amplitude) for a prolonged period of time. Moreover, in practice, clinicians often do not precisely know which facets of similarity they are interested in. Thus, it would be highly beneficial to provide them a fast means of measuring similarity capable of integrating multiple perspectives, i.e., distance functions.

The current limit of LSH is that it can hash the data with only one family of locality-sensitive hash functions for a given distance function at a time. In order to explore the data with more diverse and integrated perspectives, in [4], we introduced Stratified LSH (SLSH) for the retrieval task, which incorporates hash families for multiple distance functions, where *a*) the outer level LSH first stratifies the data by amplitude using the l_1 distance, and then *b*) hierarchically, within each strata, the inner level LSH with the cosine distance hashes the data according to angle and shape of time series. In this work, we extend SLSH to the problem of predicting a critical event, namely acute hypotensive episode (AHE), with the LSH families for the l_1 distance (L1LSH) [5] and the cosine distance (COSLSH) [6]. We demonstrate SLSH on a dataset of mean arterial blood pressure extracted from the ICU physiological waveform repository of the MIMIC-II database [7] and compare SLSH to the standard LSH and the linear KNN method. To our best knowledge, SLSH is the first practical application of multi-level LSH for the prediction task of a critical event based on physiological time series referencing a repository with thousands of patients.

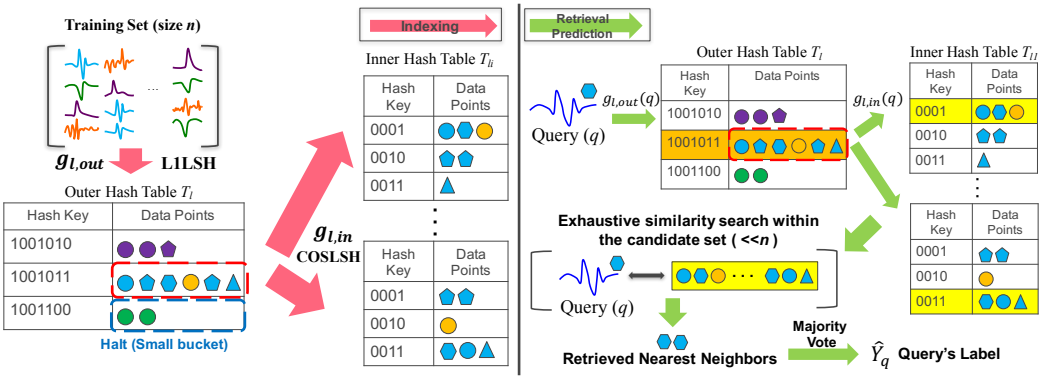

Figure 1: (*Left*) SLSH Indexing. We first stratify the data according to L1LSH at the outer level. Then, on each bucket with a significant size, we apply another layer of LSH with COSLSH at the inner level. (*Right*) SLSH Retrieval and Prediction. We retrieve the approximate nearest neighbors of a query of interest by applying the same outer and inner hash functions used for construction and perform the linear search within the candidate set. Prediction is done by the majority vote. The figure illustrates the simple case when one hash table is built at the outer level.

2 Method: Stratified Locality-Sensitive Hashing

We build the LSH based predictors for the occurrence of AHE within an event window. We train the models with a lag time amount of historical data prior to the window, where each data point has a label indicating the occurrence of AHE. The basis for the prediction by SLSH is the standard LSH [2], which consists of three steps: *a*) constructing an efficient data structure (hash table) to *index* (hash) the data for fast retrieval, *b*) quickly *retrieving* the approximate NNs of the query of interest, and *c*) *predicting* the label of the query based on predominance of labels of its NNs.

We explain the steps of indexing and retrieval more specifically as they serve as the cores of the standard LSH. For indexing, we select m functions such that $g_l = (h_{1,l}, h_{2,l}, \ldots, h_{m,l})$ for each $l = [1, 2, \ldots, L]$. The hash functions h's are randomly chosen from a LSH family H. Then, we construct L independent hash tables where each hash table T_l contains the dataset points hashed using the function g_l. The value of g_l for each data point defines its hash key. For each locality-sensitive hash function, the probability of hashing to the same value (*collision*) is much higher for points close in high-dimensional space than those that are far away. This distance preserving property is what distinguishes LSH from the conventional hashing as the goal of the latter is to avoid collisions even for close points. There are two LSH parameters, the number of tables constructed L and the number of hash functions m used per table, which need to be chosen empirically. The optimal pair of (m, L) provides the most efficient data structure to index the data for the fastest and the most accurate retrieval and prediction.

For retrieval by the standard LSH, we 1) hash a query by the same set of hash functions g_l's used for indexing, and 2) retrieve a group of k waveforms that are most similar to the query by the *linear search* within the candidate set. The candidate set is defined as the union of all points contained in the colliding hash buckets of the query from each hash table. The step 2) is the bottleneck of LSH, and where SLSH differs.

The procedures of our multi-level SLSH (Figure 1) are composed of the following tasks built on top of the standard LSH. 1) We first stratify the data according to L1LSH at the outer level with (m_{out}, L_{out}). 2) Only on each bucket/stratum with a significant size ("populous bucket", whose median size is larger than α% of the original data size), we probe deeper with COSLSH at the inner level with (m_{in}, L_{in}). This is somewhat analogous to performing top-down hierarchical clustering. 3) We retrieve the approximate NNs of a query of interest by applying the identical outer and inner (only when needed) hash functions used for construction, and by performing the linear search within the candidate set. 4) Finally, the prediction part of SLSH is done by taking the majority vote among the retrieved k approximate NNs, identical to the step c) of the standard LSH above. While we use L1LSH and COSLSH for the outer and inner level LSH, respectively, it is important to note that our SLSH framework can embrace any distance function which has a valid locality-sensitive hash family. For more detailed algorithmic procedures of SLSH, refer to [4].

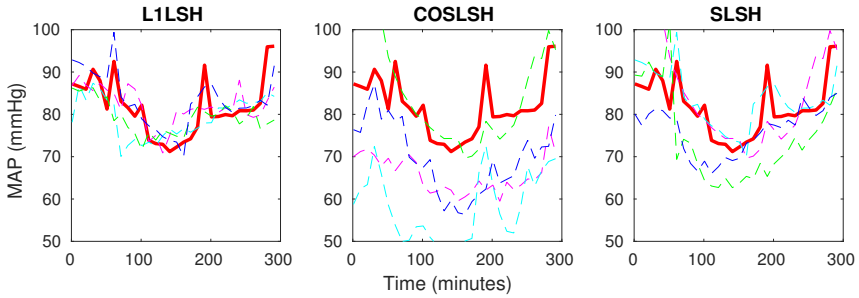

Figure 2: The 5-nearest neighbors (colored dashed lines) of a MAP waveform query (red line) retrieved by (*Left*) the standard L1LSH, (*Middle*) the standard COSLSH, and (*Right*) SLSH. Each set is similar to the query in terms of amplitude, shape, and *both* amplitude *and* shape by L1LSH, COSLSH, and SLSH, respectively.

3 Experiment and Discussion

We evaluate whether the prediction by SLSH is advantageous and by how much, compared to the single-level standard LSH with L1LSH. We present our findings on our time series dataset of mean arterial blood pressure (MAP) extracted from the large ICU physiological waveform repository of MIMIC-II (Multi-parameter Intelligent Monitoring in Intensive Care) database [7]. AHE is defined as any 30 minute interval where more than 90% of the per-minute MAP values are below 60 *mmHg*. Our dataset contains 6,467 segments of 300 minutes long (lag time) per-minute MAP from 2,291 patients and is unbalanced with 1:11 ratio between AHE positives (476 segments) and negatives (5,991 segments). We use 80%:20% of data for training and testing. The prediction accuracy is defined as $Accuracy = (TP + TN)/(TP + FP + TN + FN)$[1]. The corresponding speed-up factor is the average time taken by a kind of LSH relative to that by KNN.

3.1 Qualitative Evaluation: Retrieved Nearest Neighbor Set

We show that SLSH is capable of retrieving NNs according to both amplitude and shape by a visual example. Figure 2 illustrates the retrieved nearest neighbor sets of a MAP time series query using the standard single-level L1LSH, the standard single-level COSLSH, and SLSH. For the given query (Segment ID 6404, red line), in Figure 2 (*Left*), we observe that the set of 5-NNs (colored dashed lines) retrieved by L1LSH is tightly located close to the query in terms of the mean amplitude but oblivious to their various shapes. Likewise, in Figure 2 (*Middle*), the NNs retrieved by COSLSH all resemble the shape of the query, but their amplitudes are well-spread across various levels. Figure 2 (*Right*) shows a qualitative evaluation of SLSH. The NNs obtained via SLSH not only have shapes that are similar to that of the query, but also have their mean amplitudes much closer the query compared to the set retrieved by COSLSH. Satisfying the notion of similarity in terms of both amplitude *and* shape, this qualitatively verifies that SLSH is able to address the data from multiple and more integrated perspectives.

3.2 Quantitative Evaluation: AHE Prediction

We empirically demonstrate that the prediction by SLSH is faster and more accurate than L1LSH. Figure 3 shows the accuracy and the associated speed-up factor of L1LSH (red, green) and SLSH (blue) for the AHE prediction based on 1-NN. First, for each combination (m, L) of LSH parameters $m \in [5, 10, \ldots, 50]$ and $L \in [10, 20, \ldots, 100]$, we make a prediction via the single-level L1LSH. The prediction results with its corresponding speed-ups are shown as the red crosses where each point corresponds to a single parameter instance of (m, L). The leftmost point of the red plot with no speed-up shows the accuracy of KNN (the baseline). Then, by setting a particular instance (m^*, L^*) of L1LSH as (m_{out}, L_{out}) and $\alpha = 1\%$, we perform SLSH with the parameters $m_{in} \in [1, 4, \ldots, 19]$ and $L_{in} \in [1, 4, \ldots, 10]$. For illustration, we choose $(m^*, L^*) = (m_{out}, L_{out}) = (35, 20)$, the instance of L1LSH parameters which outputs 1% loss of accuracy from KNN with a speed-up gain of 25x (corresponding to the green point on the figure). For the entire range of SLSH (blue points),

[1] TP = True Positive, FP = False Positive, FN = False Negative, TN = True Negative

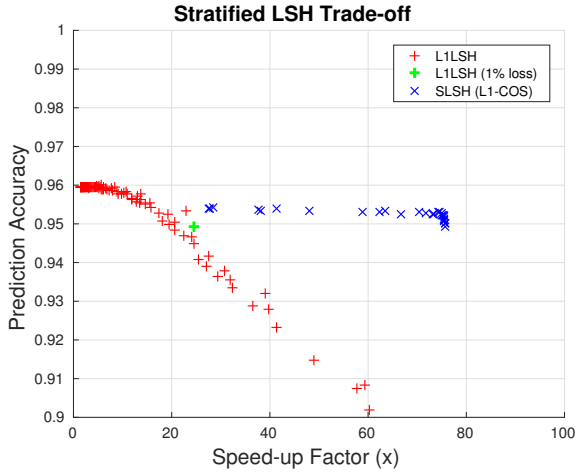

Figure 3: Comparison of SLSH to L1LSH for prediction of AHE based on 1-NN. Given a L1LSH instance (green) as its benchmark and (m_{out}, L_{out}), SLSH (blue) outperforms for the entire range of (m_{in}, L_{in}). Each point corresponds to a parameter configuration (m, L) and (m_{in}, L_{in}) of L1LSH and SLSH, respectively.

we observe that it is more accurate *and* faster than its benchmark, the single-level L1LSH (the green point). This holds true for other values of $(m^*, L^*) = (m_{out}, L_{out})$ as well.

All in all, the experiment shows that with a carefully tuned set of parameters, SLSH has a higher accuracy and a much faster querying speed than the standard LSH with L1LSH alone. For accuracy, we observe that obtaining a set of NNs from multiple perspectives indeed leads to a better prediction. We achieve a large speed up gain with SLSH by avoiding the bottleneck of L1LSH because the candidate set of SLSH subject to the linear search is orders of magnitude smaller than that of L1LSH. Also, the additional time taken to apply the inner level hash functions in SLSH is much shorter than the time required to conduct the linear search in populous buckets in L1LSH.

4 Future Work

SLSH is neither associative nor commutative. Varying the order of operations generates different results. We are particularly interested in investigating the impact of altering the order of hash families, and in examining how accuracy and interpretability change as a result. We are currently comparing the result of SLSH to L1LSH, COSLSH, and the combination of the two. In general, the larger a dataset is, the more dramatic the effect of LSH will be due to its sublinear time complexity. Thus, we plan to apply SLSH on much larger datasets with a higher resolution.

References

[1] Y. B. Kim and U.-M. O'Reilly, "Large-scale physiological waveform retrieval via locality-sensitive hashing," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 5829–5833, IEEE, 2015.

[2] Y. B. Kim and U.-M. O'Reilly, "Analysis of locality-sensitive hashing for fast critical event prediction on physiological time series," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pp. 783–787, IEEE, 2016.

[3] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of Computing*, pp. 604–613, ACM, 1998.

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

[5] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pp. 518–529, Morgan Kaufmann Publishers Inc., 1999.

[6] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of Computing*, pp. 380–388, ACM, 2002.

[7] M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark, "Multiparameter intelligent monitoring in intensive care II (MIMIC-II): A public-access intensive care unit database," *Critical Care Medicine*, vol. 39, no. 5, p. 952, 2011.