

Efficient algorithms for sequence segmentation

Evimaria Terzi * Panayiotis Tsaparas †

Abstract

The sequence segmentation problem asks for a partition of the sequence into k non-overlapping segments that cover all data points such that each segment is as homogeneous as possible. This problem can be solved optimally using dynamic programming in $O(n^2k)$ time, where n is the length of the sequence. Given that sequences in practice are too long, a quadratic algorithm is not an adequately fast solution. Here, we present an alternative constant-factor approximation algorithm with running time $O(n^{4/3}k^{5/3})$. We call this algorithm the DNS algorithm. We also consider the recursive application of the DNS algorithm, that results in a faster algorithm ($O(n \log \log n)$ running time) with $O(\log n)$ approximation factor, and study the accuracy/efficiency tradeoff. Extensive experimental results show that these algorithms outperform other widely-used heuristics. The same algorithms can speed up solutions for other variants of the basic segmentation problem while maintaining constant their approximation factors. Our techniques can also be used in a streaming setting, with sublinear memory requirements.

1 Introduction

Recently, there has been an increasing interest in the data-mining community for mining sequential data. This is due to the existence of abundance of sequential datasets that are available for analysis, arising from applications in telecommunications, stock-market analysis, bioinformatics, text processing, click-stream mining and many more. The main problem associated with the analysis of these datasets is that they consist of huge number of data points. The analysis of such data requires efficient and scalable algorithms.

A central problem related to time-series analysis is the construction of a compressed and concise representation of the data, so that it is handled efficiently. One commonly used such representation is the *piecewise-constant* approximation. A piecewise-constant representation approximates a time series T of length n using k non-overlapping and contiguous segments that span the whole sequence. Each segment is represented by a single (constant) point, e.g., the mean of the points in the segment. We call this point the *representative* of the segment, since it represents the points in the segment.

The error in this approximate representation is measured using some *error function*, e.g. the sum of squares. Different error functions may be used depending on the application. Given an error function, the goal is to find the segmentation of the sequence and the corresponding representatives that minimize the error in the representation of the underlying data. We call this problem a *segmentation problem*. Segmentation problems, particularly for multivariate time series, arise in many data mining applications, including bioinformatics [5, 15, 17] and context-aware systems [10].

This basic version of the sequence-segmentation problem can be solved optimally in time $O(n^2k)$ using dynamic programming [3], where n is the length of the sequence and k the number of segments. This quadratic algorithm, though optimal, is not satisfactory for data-mining applications where n is usually very large. In practice, faster heuristics are used. Though the latter are usually faster ($O(n \log n)$ or $O(n)$), there are no guarantees on the quality of the solutions they produce.

In this paper, we present a new *divide and segment* (DNS) algorithm for the sequence segmentation problem. The DNS algorithm has sub-quadratic running time, $O(n^{4/3}k^{5/3})$, and it is a 3-approximation algorithm for the segmentation problem. That is, the error of the segmentation it produces is provably no more than 3 times that of the optimal segmentation. Additionally, we explore several more efficient variants of the algorithm and we quantify the accuracy/efficiency tradeoff. More specifically, we define a variant that runs in time $O(n \log \log n)$ and has an $O(\log n)$ approximation ratio. All algorithms can be made to use sub-linear amount of memory, making them applicable to the case that the data needs to be processed in a streaming fashion. We also propose an algorithm that requires logarithmic space, and linear time, albeit, with no approximation guarantees.

Extensive experiments on both real and synthetic datasets demonstrate that in practice our algorithms perform significantly better than the worst-case theoretical upper bounds. It is often the case that the more efficient variants of our algorithms are the ones that produce the best results, even though they are inferior in theory. In many cases our algorithms give results equivalent to the optimal algorithm. We also compare our algorithms against different popular heuristics that are known to work well in practice. Although these heuristics output results of good quality our algorithms still

*HIIT, Basic Research Unit Department of Computer Science University of Helsinki, Finland, email:terzi@cs.helsinki.fi

†HIIT, Basic Research Unit Department of Computer Science University of Helsinki, Finland, email:tsaparas@cs.helsinki.fi

