Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®
ELSEVIER
Mathematical and Computer Modelling 39 (2004) 209–217
MATHEMATICAL AND COMPUTER MODELLING
www.elsevier.com/locate/mcm

Text Segmentation by Product Partition Models and Dynamic Programming

A. KEHAGIAS
Department of Math., Phys. and Comp. Sciences
Faculty of Engineering, Aristotle University of Thessaloniki
Thessaloniki, Greece

A. NICOLAOU
Department of Business Administration
University of Macedonia, Thessaloniki, Greece

V. PETRIDIS AND P. FRAGKOU
Department of Electrical and Computer Engineering
Faculty of Engineering, Aristotle University of Thessaloniki
Thessaloniki, Greece

(Received November 2002; revised and accepted June 2003)

Abstract—In this paper, we use Barry and Hartigan's *Product Partition Models* to formulate text segmentation as an optimization problem, which we solve by a fast dynamic programming algorithm. We test the algorithm on Choi's segmentation benchmark and achieve the best segmentation results so far reported in the literature.

Keywords—Text segmentation, Dynamic programming, Product partition models.

1. INTRODUCTION

Text segmentation is a problem of great practical significance. The goal is to divide a text into *homogeneous segments*, so that each segment deals with a particular subject while contiguous segments deal with different subjects. In this manner, documents relevant to a query can be retrieved from a large database of unformatted (or loosely formatted) text. For an overview of the problem and various methods for its solution, see [1–10].

A *product partition model* (PPM) is a Bayesian inference procedure for segmentation of a sequence of random variables, based on the *heterogeneity* of the sequence. PPMs were introduced by Barry and Hartigan [11,12] (see also [13,14]) to identify multiple *change points* in the mean and variance of a sequence of normally distributed random variables. The model assumes that the random segmentation produced by the change points has a probability distribution proportional to a product of *prior cohesions*, one for each segment. Given the observations, a new product partition model holds, with *posterior* cohesions for the segments.

0895-7177/04/$ - see front matter
doi: 10.1016/S0895-7177(04)00010-X

In this paper, we use the PPM framework to identify text segments. To this end, we obtain the posterior joint probability of an observed text and its segmentation as a product of two terms:

(a) the probability of the segmentation, described by appropriate prior cohesions, and
(b) the conditional (given the segmentation) probability of the *sentence similarity matrix*, described by an appropriate homogeneity function.

Note that we use PPMs to assign probabilities to *two-dimensional* structures (the sentence similarity matrices) rather than to one-dimensional sequences. The negative logarithm of the joint probability is the *segmentation cost*, which is minimized by a fast dynamic programming algorithm (compare to the use of computationally demanding Markov chain Monte Carlo algorithms in [11,12]). The homogeneity function we use depends on some parameters which are estimated from training data; as far as we know, this approach has not been previously used in conjunction with PPMs.

In Section 2, we describe the PPMs we use to tackle the text segmentation problem. In Section 3, we present a dynamic programming algorithm to solve the problem. In Section 4, we present some experiments to evaluate our algorithm. Finally, in Section 5, we discuss our results, review some work related to our own, and present future research directions.

2. PROBLEM FORMULATION

2.1. Representation

Consider a text with T sentences. A *segmentation* of the text is a partition of $\{1, 2, \ldots, T\}$ into K contiguous *segments*: $\{1, 2, \ldots, t_1\}$, $\{t_1 + 1, t_1 + 2, \ldots, t_2\}$, $\ldots$, $\{t_{K-1} + 1, t_{K-1} + 2, \ldots, T\}$; and $t_0, t_1, \ldots, t_K$ are the *segment boundaries*[1] which satisfy

$$0 = t_0 < t_1 < \cdots < t_{K-1} < t_K = T.$$

A concise representation of the segmentation is given by the vector $\mathbf{t} = (t_0, t_1, \ldots, t_K)$; note that vector length K (i.e., the number of segments) is variable, but it satisfies $K \leq T$. We will denote the set of all possible segmentations of $\{1, 2, \ldots, T\}$ by Φ_T.

Assume that the text has a vocabulary of L distinct words (common *uninformative* words such as "and", "the", etc., are not included). The text can be represented by a $T \times L$ matrix $\mathbf{c}$, where (for $t = 1, 2, \ldots, T$ and $l = 1, 2, \ldots, L$)

$$c_{t,l} = \begin{cases} 1, & \text{iff the } l^{\text{th}} \text{ word appears in the } t^{\text{th}} \text{ sentence,} \\ 0, & \text{else.} \end{cases}$$

The *sentence similarity matrix* of the text is a $T \times T$ matrix $\mathbf{d}$, where

$$d_{t,t} = 0, \quad \text{for } 1 \leq t \leq T, \qquad \text{and} \qquad d_{s,t} = \begin{cases} 1, & \text{if } \sum_{l=1}^{L} c_{s,l} c_{t,l} > 0, \\ 0, & \text{if } \sum_{l=1}^{L} c_{s,l} c_{t,l} = 0, \end{cases} \quad \text{for } 1 \leq s \neq t \leq T.$$

In other words, $d_{s,t} = 1$ when the s^{th} and t^{th} sentence have at least one word in common (but note that the diagonal elements $d_{t,t}$ are set equal to zero). We will denote the set of all possible sentence similarity matrices (of dimension $T \times T$) by Ψ_T. The resulting matrix $\mathbf{d}$ has zeros and ones arranged in a characteristic pattern which corresponds to the structure of the text. In

[1] We assume that segment boundaries always appear at the ends of sentences.