Estimating the number of segments in time series data using permutation tests

Kari T. Vasko & Hannu T.T. Toivonen*
University of Helsinki
Department of Computer Science
PO Box 26, FIN-00014, Finland
{Kari.Vasko,Hannu.Toivonen}@cs.helsinki.fi

Abstract

Segmentation is a popular technique for discovering structure in time series data. We address the largely open problem of estimating the number of segments that can be reliably discovered. We introduce a novel method for the problem, called Pete. Pete is based on permutation testing.

The problem is an instance of model (dimension) selection. The proposed method analyzes the possible overfit of a model to the available data rather than uses a term for penalizing model complexity. In this respect the approach is more similar to cross-validation than regularization based techniques (e.g., AIC, BIC, MDL, MML). Further, the method produces a p value for each increase in the number of segments. This gives the user an overview of the statistical significance of the segmentations. We evaluate the performance of the proposed method using both synthetic and real time series data. The experiments show that permutation testing gives realistic results about the number of reliably identifiable segments and that it compares favorably with the Monte Carlo cross-validation (MCCV) and commonly used BIC criteria.

1. Introduction

Time series segmentation is an instance of clustering analysis. It addresses the following data mining problem: given a time series T, find a partitioning of T to segments that are internally homogeneous. Depending on the application, the goal could be to locate stable periods of time, to identify change points, or to simply compress the original time series into a more compact presentation. In this paper, we are concerned with the discovery of interesting features in data rather than with compression as such. For an overview to methods and approaches to time series segmentation see, e.g., [7, 8, 13, 5, 6].

As an example application, consider the analysis of biostratigraphic data collected from microfossils accumulated and preserved in lake sediments [4, 11]. Paleoecologists obtain deep cores of sediment and analyse them layer by layer. Deeper layers correspond to distant points in time and the topmost layers are the most recent ones. Typically such data spans a period from hundreds to thousands of years. In each layer (time point), the species composition of some suitable organisms is analysed. The abundances of these species in the different sediment layers form a multidimensional time series, where homogeneous segments correspond to environmentally relatively stable periods and segment boundaries to more significant changes in the environment.

We introduce a novel method, *Pete*, for estimating the number of segments that can be reliably found in a given time series. The method is based on permutation tests on the available data. Instead of defining a data independent measure for model complexity, like penalized likelihood based approaches, it only measures the dataset specific overfit resulting from increased model complexity, and is in this respect similar to cross-validation techniques.

Another recent application for time series segmentation is in context sensitivity of mobile devices. Being able to measure and sense the environment is not enough: a problem that remains is how to recognize different contexts from the measurements. An approach proposed recently is to segment the measurement time series and, in the spirit of unsupervised learning, to identify different contexts with the discovered homogeneous segments [6]. Again, the number of segments to be discovered has a significant effect on the results.

In a nutshell, the Pete algorithm can be described as follows. The given time series is segmented in steps to $m = 1, 2, \ldots$ segments. The segmentation can be done with any segmentation algorithm; all that Pete needs from each segmentation is the amount of error that remains (or the goodness of fit). Pete recognizes overfitting by analyzing at each segmentation step m the reduction of error. The relative reduction is contrasted to the respective reduction

* Also at Nokia Research Center

in the case there no segment structure in the data. If the reduction is not significantly better in the observed data then segmentation is stopped.

2. Background and definitions

2.1. Time series and segmentations

A time series $T = (x(t)|1 \leq t \leq n)$ is a finite set of n samples labeled by time points $1, 2, \ldots, n$. A segment of T is a set of consecutive time points $S_T(a, b) = (t|a \leq t \leq b)$. An m-segmentation S_T^m ($m \leq n$) of time series T is a partition of T to m non-overlapping segments

$$S_T^m = \{S_T(a_i, b_i)|1 \leq i \leq m\}$$

such that $a_1 = 1, b_m = n$, and $a_i = b_{i-1}+1$ for $1 < i \leq m$. In other words, an m-segmentation splits T to m disjoint time intervals. For simplicity, the segments are denoted by $S_1, \ldots, S_m$.

Usually the goal is to find homogeneous segments from a given time series. For instance, in paleoecological studies an objective is to find periods of time where the species composition and thus also the climate has been relatively stable. In such a case the segmentation problem can be described as constrained clustering: data points should be grouped by their similarity, but with the constraint that all points in a cluster must come from successive time points.

We assume that the segmentation is based on fitting a (simple) function within each segment, and on searching for a segmentation that results in a good overall fit. Usually the function is a constant or linear function, or a polynomial of a higher but limited degree, fitted to approximate the values within the segment.

2.2. The error of a segmentation

A good segmentation fits the data well and has a small error. The estimation error is usually defined as a positive function of the distances between the actual values in the time series and the values given by the functions within segments.

Let T be a time series of length n and $S_T^m = \{S_1, \ldots, S_m\} \in \mathcal{S}_m$ an m-segmentation, where $\mathcal{S}_m$ denotes the set of all possible m-segmentations of T. Let θ_i denote the vector of parameters for the function in segment S_i. (For a constant function there obviously is only one parameter and for a non-continuous linear function there are two parameters per segment.)

The error

$$e(S_T^m) = e(S_T^m, \theta_1, \theta_2, \ldots, \theta_m) \tag{1}$$

of a segmentation describes how far the model, i.e., the function consisting of the local functions in the segments, is from the data. The error function is usually the sum of squared errors or a function of the (log) likelihood in probabilistic settings. Since different approaches base the segmentation on different error measures, we do not assume any particular segmentation algorithm or error function. We do assume that the modeling error $e(S_T^1)$, $e(S_T^2), \ldots$ for a given segment T is a positive non-increasing function of the number of segments.

2.3. Finding good segmentations

Suppose that finding the parameter values $\theta = (\theta_1, \ldots, \theta_m)$ that (approximately) minimize the error function for a given segmentation S_T^m is a tractable task, e.g., computation of the least squares line. Given that θ is easily available the question remains how to explore the set $\mathcal{S}_m$ for good solutions: the search space is exponential in n, the length of the time series.

The optimal m-segmentation can be characterized in a recursive way so that dynamic programming can be used to find it (e.g. [6]). Unfortunately, dynamic programming is computationally intractable for many real data sets. Consequently, heuristic optimization techniques such as greedy top-down or bottom-up techniques are frequently used to find good but suboptimal m-segmentations [6, 7].

In this paper, we do not address the problem of how to find good segmentations. We simply assume one of the many available methods is used.

2.4. Relative reduction of error

Recall that $e(S_T^m)$ is the error of segmentation S_T^m. Then

$$RR(m|T) = \frac{e(S_T^{m-1}) - e(S_T^m)}{e(S_T^{m-1})} \tag{2}$$

is the relative reduction of error when m segments are used instead of $m-1$ segments. The Pete method will analyze the relative reduction of error as the number of segments is increased and stop segmentation when statistically significant reductions are not achieved any more.

The assumptions we make about the underlying segmentation algorithm and the error function can be summarized as follows. For a fixed time series T and a given segmentation algorithm, the error $e(S_T^1)$, $e(S_T^2), \ldots$ is a positive non-increasing function of the number of segments. This follows naturally if the power of the local functions is constant, as is usually the case. Obviously then $RR(m|T)$ is in $[0, 1]$ for all $m > 1$.

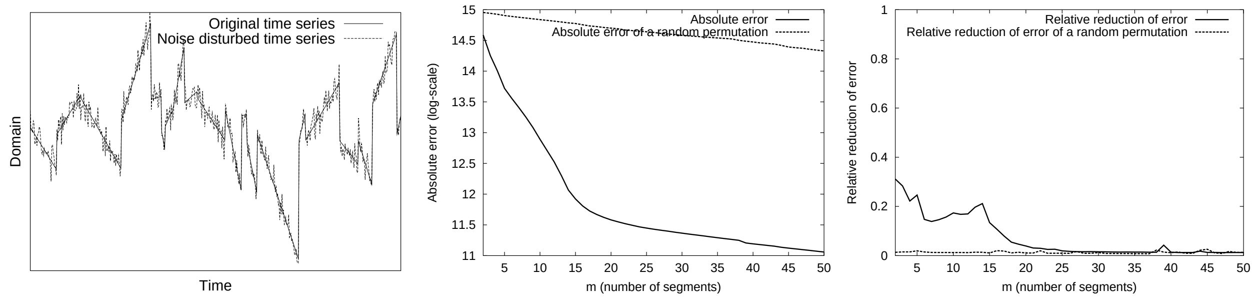

Figure 1. A noisy piece-wise linear time series (left). Absolute error (middle) and relative reduction of error (right) as a function of the number of segments.

3. Estimating the number of segments

3.1. The method

We now introduce the Pete algorithm for determining an appropriate number of segments in a given time series. The idea is perhaps best conveyed by a simple example. We will justify the use of permutation tests more formally in the following subsection.

Imagine a piecewise linear time series consisting of, say, around 20 segments, with some noise in the data (Figure 1, left). Segmenting the time series into $m = 2, 3, \ldots, 20$ segments keeps improving the fit and decreasing the error, as more and more of the true segments of the data are matched (solid line in the middle panel of Figure 1). With $m = 21, 22, \ldots$ segments the error keeps decreasing as well, but the reduction of error is due to noise in the data. We devise a method that tests if the reduction of error is likely to be due to noise or due to some structure in the data.

So how to tell if the reduction of error is due to noise? Let T be the given time series of length n, and let $\mathcal{T}$ be a similar random time series consisting of noise only and of no non-random structure. What we want to do is to test the alternative hypothesis $h_1 : RR(m|T) > RR(m|\mathcal{T})$ against the null hypothesis $h_0 : RR(m|T) = RR(m|\mathcal{T})$. If we can test the null hypothesis and the observed reduction of error is very unlikely to result from noise, then (at least) m segments can be considered justified.

Where to obtain random time series $\mathcal{T}$ that are somehow representative of the properties of the original time series T, without making any assumptions about T? The short answer is to use permutations of T: randomly permute the order of the datapoints, and the result is a random time series with value distribution identical to T. Figure 1 shows the absolute errors (middle) and their relative reductions (right) for the time series (left) and for one randomly permuted time series. For the random time series with no temporal structure the error decreases constantly — but remarkably slowly. The original and the permuted time series do not have similar reductions of error until about 22 segments.

The null hypothesis can be tested by generating a number of random timeseries $\mathcal{T}$ having the same value distribution as T and counting how often $RR(m|\mathcal{T}) \geq RR(m|T)$. This (approximated) probability of getting reduction $RR(m|T)$ by chance is the p value of the data under the null hypothesis. If it is low, e.g. below 0.05, then $RR(m|T)$ is unlikely to result from noise and m segments are justified.

Table 1 gives the Pete algorithm. To sum it up: we propose to generate random permutations of the given time series and to compare the reduction of error in the original data to that of permuted data sets, and to stop segmentation when a relatively large fraction (say, 0.05) of the permuted data sets has at least as good reduction of error as the original data set. Throughout this paper we use a cutoff value of 0.05 for the p value.

3.2. Statistical justification for the method

We next go through the approach more formally. Denote by M_T the set of all time series of length n whose value distribution is identical with the given time series T of length n. Further, let $\mathcal{T}$ be a uniformly distributed random time series in M_T.

Let $I_T(\mathcal{T}, m)$ be an indicator (a Bernoulli random variable) for whether $\mathcal{T}$ results in no smaller reduction of error than T when segmented to m segments:

$$I_T(\mathcal{T}, m) = \begin{cases} 1 & \text{if } RR(m|\mathcal{T}) \geq RR(m|T) \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Let us denote by $p_T(m)$ the probability of $RR(m|\mathcal{T}) \geq RR(m|T)$, i.e., of $I_T(\mathcal{T}, m) = 1$.

The p value of the m-segmentation of T, denoted $p_T(m)$, can be estimated as follows. First, we have

$$p_T(m) = \frac{\sum_{\mathcal{T}_i \in M_T} I_T(\mathcal{T}_i, m)}{|M_T|}. \tag{4}$$

Algorithm Pete

Input: time series T, segmentation algorithm A, error function e, number of permutations N, cutoff value p' for p value

Output: number of reliably identifiable segments

Method:

1. Generate N random permutations $\mathcal{T}_i$ of T
2. Use segmentation algorithm A to compute $e(S_T^1)$ and each $e(S_{\mathcal{T}_i}^1)$, i.e., errors with one segment
3. For $m := 2, \ldots, n$ (where n is the length of T):
   - 3.1. Use A to segment T and each $\mathcal{T}_i$ to m segments
   - 3.2. Let $r := |\{\mathcal{T}_i : RR(m|\mathcal{T}_i) \geq RR(m|T)\}|/N$ be the fraction of permutations where the reduction of error was at least as good as with T
   - 3.3. If $r > p'$ then return $m - 1$
4. Return n

Table 1. Pete permutation procedure for estimating the number of segments

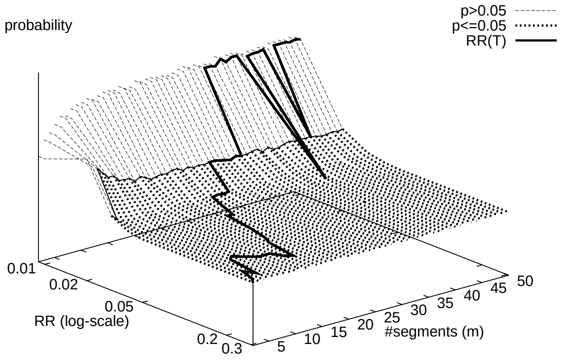

Figure 2. Monte Carlo approximated probabilities $p[RR(m|\mathcal{T}) = x|T]$of the relative reduction (RR) given the time series T of Figure 1 (left) and the number of segments.

Second, permutations can be used to approximate the right hand side of Equation 4: instead of summing over all $\mathcal{T}_i \in M_T$, we sum over a random sample from M_T, obtained using random permutations of the original time series T.

Permutation testing is a special case of Monte Carlo integration: the p value is estimated with N random points as

$$\hat{p}_T^N(m) = \frac{1}{N} \sum_{i=1}^{N} I_T(\mathcal{T}_i, m) \approx \mathbf{E}(I_T(\mathcal{T}, m)) = p_T(m) \quad (5)$$

where $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$ are uniform random samples from M_T. Random sampling from M_T can be carried out efficiently by simply generating random permutations of T, since the set M_T is exactly the same as the set of all permutations of time series T.

Figure 2 illustrates the Monte Carlo estimated distribution of $RR(m|\mathcal{T})$ ($m = 2, \ldots, 50$) as well as the relative reduction of the observed time series (Figure 1) obtained in 1000 random permutations. The region where $p \leq 0.05$, i.e., where the relative reduction is statistically significant, is drawn with stronger lines.

We emphasize that the p value $p_T(m)$ is used only as a tool to decide when to stop segmentation. It should not be thought of as a statement about the probability of any exact number of segments. Also, we obviously do not expect the data to actually consist of any constant or linear segments, even when evidence for several segments is discovered. Segmentation itself is a tool for discovering useful structure in the data, and constant and linear segments are suitable classes of concepts to be used in such an analysis.

3.3. Related work

The problem of determining an appropriate number of segments is a special case of estimating the number of dimensions of a model. The question is about overfitting: how to avoid too complex models that fit noise in addition to the 'true signal'.

There are several approaches for determining the dimension of a model. They can be roughly divided into two groups: (1) some measure model complexity without regard to the data, as well how well a model fits the data, (2) some only analyze the fit to the data.

(1) *Penalized likelihood*, e.g., MDL, MML, BIC, AIC, SIC and structural risk minimization [1, 12, 3, 16, 17], and generally all regularization based approaches to estimate an unknown target function, specify a cost for the model complexity and then minimize the sum of the model complexity and error. In the *pure Bayesian approach* the full probability model for all variables of the application domain, including model dimension m, is defined, including (data independent) prior distributions. The Bayesian approach can be considered as an instance of the penalized likelihood without asymptotically derived complexity term.

(2) *Cross-validation* is is a general and well-understood technique for addressing overfitting. In particular, *Monte Carlo cross-validation*, a variant of the standard v-fold cross-validation, has been successfully applied to model dimension selection [14, 15]. *The generalized likelihood ratio test* [2] is an example of a test on the increase in the model likelihood, compared to an assumed underlying distribution. There are techniques that try to locate a knee in the error or likelihood curve (cf. Figure 1, middle). The

broken stick model [10] is widely used by paleoecologists in the zonation problem. The broken stick model gives the expected amount of variance (error) accounted for by a segment, under the assumption that those variances follow the broken stick distribution. It is primarily applicable in top-down segmentation, where splits are added incrementally to an existing segmentation.

The permutation test method Pete differs significantly from methods in the first category since it does not assign any data-independent cost to model complexity. It does have an analogy to the generalized likelihood ratio test [2] when the error function is the log-likelihood function. However, the biggest difference to most methods in the second category is that Pete does not assume any particular distribution for the data or the error function. Cross-validation is clearly the closest match to Pete in these respects.

4. Experiments

We evaluated Pete against BIC and MCCV using both synthetic and real data. Since the real data has no "right" answers, we focus here on the synthetic cases.

4.1. Generation of synthetic data

We generated simulated data in order to assess the performance of Pete in controlled settings. The data was generated using random piecewise constant models. The performance was evaluated with respect to signal to noise ratio (SNR).

For the piecewise constant time series, the constants were generated from the normal distribution N(0,σ_{loc}), and noise was added from the normal distribution N(0, σ). The change points were drawn from a uniform distribution. In data generation, σ was a constant and the signal to noise ratio SNR $= \frac{\sigma_{loc}}{\sigma}$ was adjusted by varying the variance σ_{loc} of the signal (Figure 3). The synthetic data sets consisted of 100 data points each.

4.2. Evaluation methodology

The true number of statistically identifiable segments is not known for the data sets with noise, so we do not have any right answers to the number of segments, except for the extreme cases. Without any noise (SNR $\to \infty$) exactly the original number of (noiseless) segments should ideally be identified; and with noise alone (SNR $\to$ 0), only one segment should be discovered. Sensitivity to the original, noiseless segments was used as the basis for the numerical results, despite these shortcomings.

The following parameters were used in estimating the number of segments. Segmentations to piecewise constant functions were done with the greedy top-down method, and the error function was the sum of squared errors (which is proportional to log-likelihood function with normal noise). A cutoff value $p \leq 0.05$ was used to decide whether to continue segmentation. 2500 permutations were used to estimate the p values. Finally, all results with the synthetic data sets were averaged over 100 similarly generated random data sets.

The prediction accuracy obtained by Pete was evaluated against the penalized likelihood-based BIC-score [12] and Monte Carlo cross-validation [14, 15].

We evaluated the performance of BIC in two ways: (i) in a realistic setting, in which the variance of noise was estimated from the residuals (maximum likelihood), and (ii) in an unrealistic setting in which the true variance of noise was given as a parameter (we call this "oracle BIC" to reflect the fact that the variance is not known for real data sets). The first setting gives a practical benchmark for the performance of the method, and the second setting gives an (unfair) upper limit for the performance of BIC. For the permutation tests there obviously is no need to estimate the variance of noise.

Monte Carlo cross-validation (MCCV) or 'repeated-learning testing' was carried out as follows. Given a time series T, randomly selected 50% of time points in T are assigned to the test set, and the remaining 50% constitute the training set. This procedure is repeated M times, giving M paired test sets T_i^{test} and training sets T_i^{train} ($1 \leq i \leq M$). For each number of segments $m = 1, 2, \ldots$, the training set T_i^{test} is used for segmentation and the segmentation error is then computed in the corresponding test set T_i^{train}. In the MCCV procedure we choose the number m of segments that minimizes the average of test errors. The average is taken over several random splits M to reduce variance in the estimates.

Recent experiments indicate that MCCV might give more reliable results than v-fold cross-validation in terms of choosing the correct number of cluster components [14, 15]. We adopted in our experiments split fraction 50% since it is reasonably robust across a variety of problems [14]. The number of splits was chosen to be $M = 100$.

4.3. Results

Figure 4 gives a summary of the experimental comparisons between Pete, BIC, Monte Carlo Cross-validation and the imaginary "oracle BIC". SNR decreases and the amount of noise increases from top to bottom. The left and right columns differ only in the number of segments. Except in the very noisy bottom row, Pete and "oracle BIC" quite consistently peak at the actual number of segments and have the highest density region roughly around it. BIC, instead, tends to position its estimations systematically to smaller values, and so does MCCV. MCCV tends to make "flat"

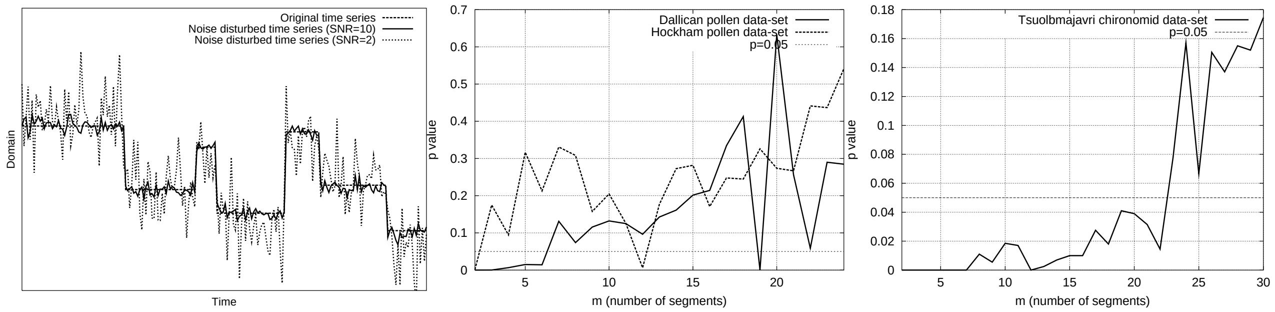

Figure 3. Two example time series consisting of 7 piecewise constant segments and two different levels of noise (SNR=2 and 10) (left panel). P values for segmentations of paleoecological data sets (middle and right panel)

predictions, i.e., the variance is large. The expected number of segments is however closer to the original number than in the results from BIC. All methods — quite expectedly — find less and less segments as the amount of noise is increased.

Based on the results, Pete is a most competitive approach. It outperforms BIC and MCCV almost constantly. Pete compares well even with the non-existing "oracle BIC" or the "BIC upper limit".

An interesting phenomena is that Pete sometimes gives very small estimates for the number of segments, even with large SNR (little noise). This is probably due to the permutation test being conservative and predicting a small number of segments if evidence for more segments is not very strong. It is also possible that a single p value above the cutoff value stops segmentation prematurely. Overall, the cutoff value obviously has a direct effect on the number of segments. We do no not believe, however, that increasing the value from 0.05 to get more segments would give better results. On the contrary: it is likely that spurious segments would then be discovered. Analysis of this is a topic for future research.

4.4. Real data

We applied Pete to three real-world data sets from the field of paleoecology. Paleoecological analysis of the first data set is presented in [9] and the two other data sets have been discussed in [4].

The first data set (Tsuolbmajavri) consists of chironomid assemblages collected from a lake sediment in northern Lappland. It is composed of 51 chironomids in 148 sediment layers [9]. The other two data sets (Dallican and Hockham)[1] are sediment cores consisting of pollen data [4]. Dallican data set consists of 23 pollen taxa in 80 time points, and Hockham data set consists of 132 pollen taxa in 163 time points. The high dimensionality with respect to the length of the time series makes segmentation of these data sets very challenging. (Obviously, dimensionality reduction techniques could be useful here, but they are outside the scope of this paper.)

Middle and right panels of Figure 3 shows the p value curves obtained for Tsuolbmajavri, Dallican and Hockham data sets, respectively (top-down algorithm, sum of variances within segments as error function, 10000 permutations). With a cutoff value $p \leq 0.05$, the p value curves can be interpreted to indicate that the Dallican data set consists of 6 segments, and that there are only two reliably identifiable segments in the Hockham data set. For the chironomid data set of Tsuolbmajavri it seems, in turn, that it is possible to identify 23 segments.

5. Conclusions and future work

We have introduced a new method, called Pete, for estimating the number of segments in time series data. Pete compares the quality of segmentations of the given data to the quality of segmentations of its random permutations.

We evaluated the performance of Pete against BIC and MCCV using synthetic time series. Experimental results show that Pete is very competitive: it was able to give very realistic estimations about the number of segments, and clearly and consistently seems to outperform BIC and MCCV in this task. However, permutations are computationally heavy and not necessarily suitable for time critical systems.

BIC was used here as a representative of a family of methods that use a model complexity term for model selection. Unlike many methods based on Occam's Razor, the proposed method does not use such a term. This is achieved by examining the derivative of the error function (relative

1. We acknowledge Keith D. Bennett and European Pollen Database for permission to use Hockham and Dallican data sets and Atte Korhola and Heikki Olander for permission to use Tsuolbmajavri data-set

reduction of error), e.g., the likelihood function, with respect to the derivates of the errors in a control group, as the model complexity is increased.

Cross-validation is the closest match to Pete in an important respect: the available data alone is used to select the model. No model complexity term nor assumptions about the data are needed. The subtle difference is that cross-validation holds out a part of the available data as a test set to estimate how well a model learned from the rest of the data actually generalizes. The permutation procedure in turn uses the full data set and its permutations to estimate the probability of obtaining a good model (in the sense of a small training error) just by chance. One reason for the inferior performace of cross-validation in our experimental results might be due to the size of the data set used to build the model: permutation tests use all of the available data whereas cross-validation holds back a fraction (50% in our experiments, selected based on results in [14]).

Possible future research issues include the following topics.

(i) The effect of different segmentation algorithms and error functions. For instance, consider greedy top-down segmentation that adds one segment boundary at a time without moving the existing ones. Instead of permuting the whole time series we could permute each segment separately, just like the top-down method tests the effect of a split in each existing segments, to obtain a p value that more closely matches the selected induction principle.

(ii) The approach we presented here is based on using a random sample from a data-dependent control group in model selection. Which kind of problems could this approach be generalized to?

(iii) Multidimensional data sets. Our current artificial data sets are one dimensional (in addition to time) but many of the interesting applications, such as the paleoecological data, are multidimensional. More tests are needed in this front.

(iv) An analysis of the p values. The longer the time series, the more p values are generated, and the more likely it is that some seem to indicate significant results, just by chance. For instance, what can be said of isolated small p values, such as the ones in two of our ecological data sets (Figure 3, middle)? Do they indicate a larger number of segments, or are they just random effects? Currently we bend towards the latter assumption.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716 – 723, 1974.

[2] F. Arnold. *Mathematical Statistics*. Prentice-Hall, New Jersey, 1990.

[3] A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

[4] K. Bennett. Determination of the number of zones in a biostratigraphical sequence. *New Phytol.*, 132:155–170, 1996.

[5] X. Ge and P. Smyth. Segmental Semi-Markov models for endpoint detection in plasma etching. Technical Report 00-08, UCI-ICS, 2000.

[6] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, and H. T. Toivonen. Time series segmentation for context recognition in mobile devices. In *The 2001 IEEE International Conference on Data Mining (ICDM'01)*, pages 203 – 210, San Jose, California, November–December 2001.

[7] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *IEEE International Conference on Data Mining*, pages 289–296, 2001.

[8] E. Keogh and P. Smyth. A probabilistic approach to fast pattern matching in time series databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97)*, pages 20–24. AAAI Press, 1997.

[9] A. Korhola, K. Vasko, H. Toivonen, and H. Olander. Holocene temperature changes in northern Fennoscandia reconstructed from chironomids using Bayesian modelling. *Quaternary Science Review*, 21:1841–1860, 2002.

[10] R. MacArthur. On the relative abundance of bird species. *Proceedings of the National Academy of Science*, 43:293 – 295, 1957.

[11] H. Mannila, H. Toivonen, A. Korhola, and H. Olander. Learning, mining, or modeling? A case study in paleoecology. In *Discovery Science, First International Conference*, Lecture Notes in Artificial Intelligence 1532, pages 12 – 24, Fukuoka, Japan, 1998. Springer-Verlag.

[12] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 7(2):461 – 464, 1978.

[13] H. Shatkay and S. B. Zdonik. Approximate queries and representations for large data sequences. In *Proceedings of the 12th International Conference on Data Engineering*, pages 536–545, Washington - Brussels - Tokyo, Feb. 1996. IEEE Computer Society.

[14] P. Smyth. Clustering using Monte Carlo cross-validation. In *Second International Conference on Knowledge Discovery and Data Mining*, pages 126–133, Portland, Oregon, USA, 1996.

[15] P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. Technical Report UCI-ICS 98-09, Information and Computer Science, University of California, Irvine, CA, 1998.

[16] M. Sugiyama and H. Ogawa. Subspace information criterion for model selection. *Neural Computation*, 13(8):1863 – 1889, 2001.

[17] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, N.Y., 1995.

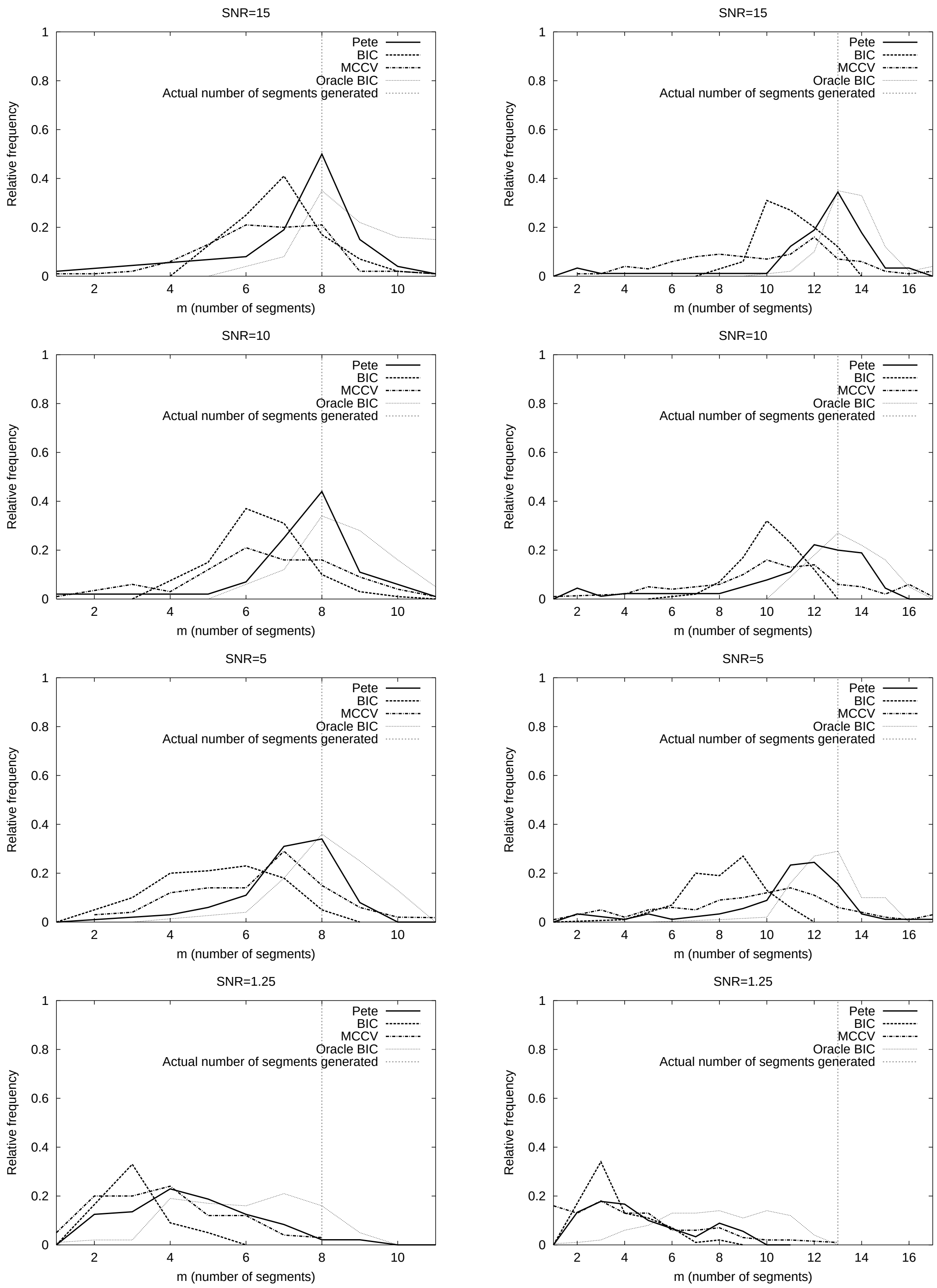

Figure 4. Distributions for the estimated number of segments obtained by the proposed permutation test (Pete), BIC, Monte Carlo cross-validation (MCCV) and "oracle BIC". The true number of segments is 8 (left column) or 13 (right column). Each graph is obtained using 100 random data sets.