Organization

- Lectures: Heikki Mannila January 21 - February 25, Fri 10-12 B222
- Email: Heikki.Mannila@cs.helsinki.fi, phone (09) 191 51246.
- Room: Exactum A339 (3rd floor)
- Course assistant: Taneli Mielikäinen, A343
- Course web page: http://www.cs.helsinki.fi/group/aprill/asp/

Organization (cont.)

- Exam: Take-home examination; questions given on February 25, answers due March 18. About 5 questions.
- (Probably) Some homework assignments during the course; completed homework counts as 30–40 % of the take-home exam. (I.e., you don't have to answer all 5 questions.)

582471 Algorithms for Segmentation Problems (2 cu)

- Lots of sequences and time series around
- Trying to understand the process that produces the sequences
- Find homogeneous pieces
- Or find pieces that are easy to describe (e.g., piecewise linear)
- Algorithmic techniques
- Probabilistic modeling
- Some applications

Tentative contents

- Introduction: sequences and sequence segmentation; what and why
- Basic problem definition and the basic dynamic programming algorithm
- Experimental studies, applications
- Probabilistic versions of the problem
- Approximate solutions
- Algorithmic changes: top-down and iterative methods for segmentation
- Finding recurrent sources: (k, h)-segmentation
- Markov chain Monte Carlo methods

1. Introduction

- Types of sequences
- Examples
- Sequence segmentation: problem definition

Types of sequences

- Strings: $a_1 a_2 \cdots a_n$, where $a_i \in \Sigma$
- Time series: $a_1, a_2, \ldots, a_n$, where $a_i \in R$
- Point process / sequence of events: $(t_1, t_2, \ldots, t_n)$ (the times in which an event happens
- Marked point process / sequence of events: $((e_1, t_1), (e_2, t_2), \ldots, (e_n, t_n)$
  - $e_i \in \Sigma$ (set of possible event types)
  - $t_i \in R$ (occurrence times)

Material

- Course is based on a selection of papers
- No textbook
- Slides will be made available
- In many cases not very long before the lecture

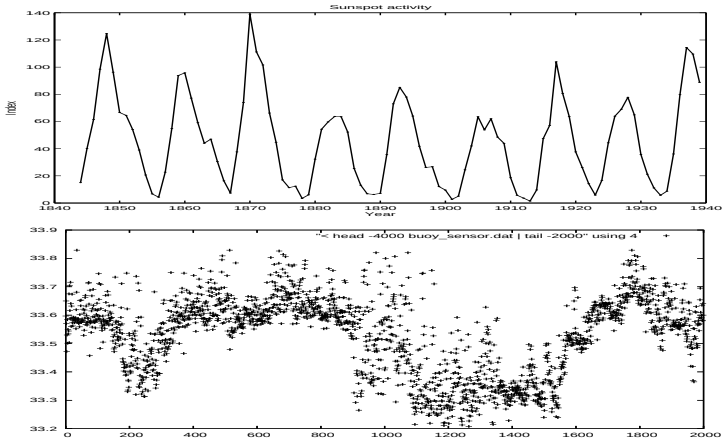
A list of concepts

Mark each of the following concepts on the scale of 1 (never heard) to 4 (very familiar)

	1	2	3	4
Random variable				
Expectation				
Geometric distribution				
Normal distribution				
Beta distribution				
Bayesian statistics				
MCMC				
L_2 metric				
L_p metric				
MDL				
Entropy of a distribution				
Log-likelihood				
Maximum likelihood				
p-value				

	1	2	3	4
Dynamic programming				
Edit distance				
Streaming algorithms				
O-notation				
NP-hardness				
Approximation algorithm				
Divide-and-conquer				
Time series				
Sequences of events				
Episodes				
Rule discovery				
Clustering				
k-means algorithm				
Hierarchical clustering				
EM-algorithm				
Bayes networks				

Examples of time series



Financial time series

Process monitoring

Examples of sequences of events

- telecommunication alarms
- biostatistical events

7406 13:17:11
7410 13:17:31
7406 13:17:41
7410 13:28:31
7406 13:28:41
7001 13:28:57
7410 13:40:11
7406 13:40:21
7001 13:48:46
7410 13:50:00
7406 13:50:10
7001 14:42:25

All data is sequential :-)

- Why?
- All data items arrive in the data store in some order
- Examples
  - Transaction data
  - Documents and words
- In some or many cases the order does not matter
- In many cases it is of interest

Examples of strings

Documents

DNA

tataacacaaataataaatgcttgaggggatgaatatccaattttccatt
atgtacttattgtacattgcatgcctgtaccaaaatatttcatgtacctc
ataaatgtatacacctgctatgtacccacaaaaattaaattttaaaacaa
tacattgttatccactatagtcaccatattgcacaatagatctgttgaat
tcattcctcctgtacaatgcaattttgtacccctttgaccaacatctaccc
aatcctcctggtaaccatcattctactctgtacttctatgtgttcagcct
tcttagacctccacatacaagtgagattatgcagtatctggctttctgtg
cctggattattttactcagtataatgtcctcccggttcattcatgttgtc
acaaatgatacttttttattttttaaggttgtatactattctattgtgt
atgtgtaccacattttcttcatccactcatgtgtcgatggatacttaagt
taattccacatcttggctgttgtgaataatgctacaataaatatgggagt
acagataactcattgacacactgatttgatatctttttaatatatgccca

H. Imai and M. Iri. *An optimal algorithm for approximating a piecewise linear function*. Journal of Information Processing, 1986.

E. Keogh and M. Pazzani. *An enhanced representation of time series that allows fast and accurate classification, clustering and relevance feedback*. International Conference on Knowledge Discovery in Data and Data Mining, 1998.

H.J.L.M. Vullings, M.H.G. Verhaegen, and H.B. Verbruggen. *ECG segmentation using time-warping*. International Symposium on Intelligent Data Analysis, 1997.

H. Shatkay and S. Zdonik. *Approximate queries and representations for large data sequences*. International Conference on Data Engineering, 1996.

C. Wang and X. Wang. *Supporting content-based searches on time series via approximation*. In proceedings of the 12th International Conference on

B. Yi and C. Faloutsos. *Fast time sequence indexing for arbitrary L_p norms*. International Conference on Very Large Data Bases, 2000.

E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. *Locally adaptive dimensionality reduction for indexing large time series databases*. International Conference on Management of Data, 2001.

V. Guralnik and J. Srivastava. *Event detection from time series data,* International Conference on Knowledge Discovery in Data and Data Mining, 1999.

H. Mannila and M. Salmenkivi. *Finding simple intensity descriptions from event sequence data*. International Conference on Knowledge Discovery in Data and Data Mining, 2001.

Sequence segmentation

- Find structure from sequences
- What is structure?
- In sequence segmentation: structure = homogeneous segments
- Can the sequence $S = (a_1, a_2, \ldots, a_n)$ be described as a concatenation of subsequences $S_1, S_2, \ldots, S_k$ such that each S_i is in some sense homogeneous?
- What does homogeneous mean?
- *Simple to describe*
- Can be generated by a simple mechanism
- E.g, constant + some noise

References

R. Bellman. *On the approximation of curves by line segments using dynamic programming*. Communications of ACM, 1961.

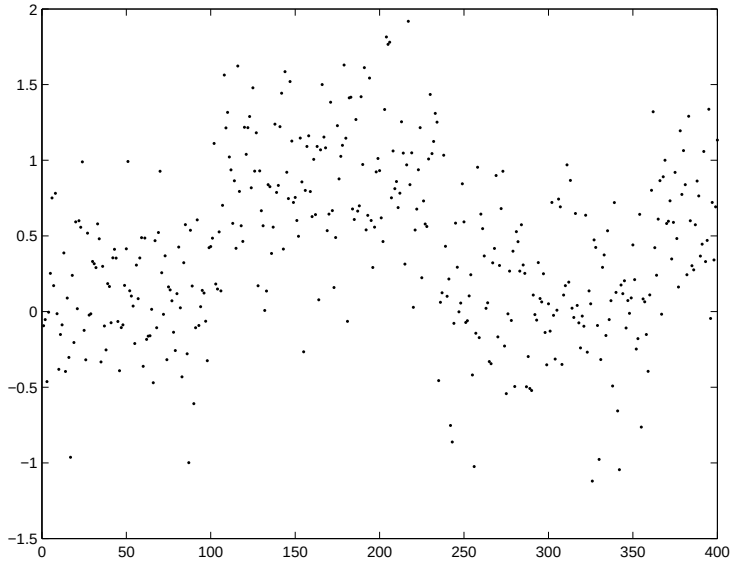
A. Cantoni. *Optimal curve fitting by piecewise linear functions*. IEEE Transactions on Computers, 1971.

T. Pavlidis and S.L. Horowitz. *Segmentation of plane curves*. IEEE Transactions on Computers, 1974.

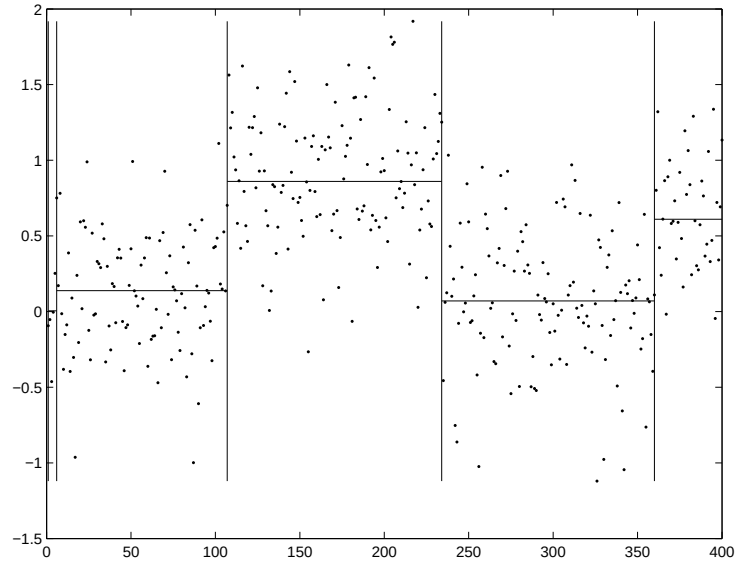
T. Pavlidis. *Waveform segmentation through functional approximation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1980.

L.D. Wu. *Piecewise Linear approximation based on a statistical model*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984.

Example: a time series with 400 points



Generating segment structure of the sequence



J. Kleinberg. *Bursty and hierarchical structure in streams*. International Conference on Knowledge Discovery in Data and Data Mining, 2002.

A. Gionis and H. Mannila. *Finding recurrent sources in sequences*. International Conference on Research in Computational Molecular Biology, 2003.

J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki and H. Toivonen. *Time-series segmentation for context recognition in mobile devices*. IEEE Conference on Data Mining 2001, p. 203-207.

W. Li. *DNA segmentation as a model selection process*. International Conference on Research in Computational Molecular Biology, 2001.

R. Azad, J. Rao, W. Li, and R. Ramaswamy. *Simplifying the mosaic description of DNA sequences*. Physical Review E, 66, 2002.

M. Koivisto, et al. *An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries*. Pacific Symposium on Biocomputing, 2003.

X. Ge, W. Pratt, and P. Smyth. *Discovering Chinese words from unsegmented text*. Research and Development in Information Retrieval, 1999.

H.J. Bussemaker, H. Li, and E.D. Siggia. *Regulatory element detection using a probabilistic segmentation model*. Intelligent Systems for Molecular Biology, 2000.

E. Agichtein and V. Ganti. *Mining reference tables for automatic text segmentation.*

International Conference on Knowledge Discovery in Data and Data Mining, 2004.

L_p metrics

- $p > 0$
- L_p norm $L_p(B) = ||b||_p$ of a vector $b = (b_1, \ldots, b_m)$:

$$\left(\sum_{i=1}^{m} b_i^p \right)^{1/p}$$

- Distance $L_p(x, y) = L_p(|x - y|)$
- $p = 2$: Euclidean metric
- $p = 1$: Manhattan metric
- $p \to \infty$: maximum metric
- What does the ball $\{x|\ L_p(x) = 1\}$ look like?
- L_p^p norm: $L_p^p(b) = \sum_{i=1}^{m} b_i^p$

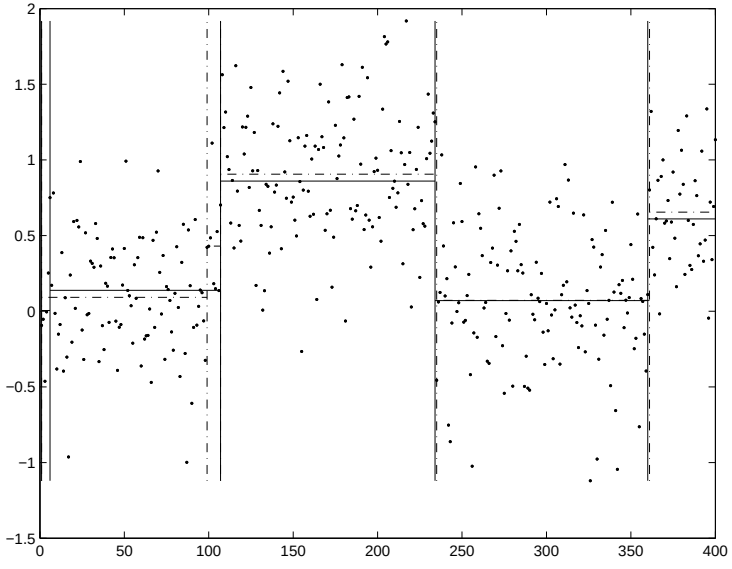
Piecewise constant modeling of a time series

- Simple description = "the values of the time series in a segment are constant"
- Constants $c_1, \ldots, c_k$
- $a_i \approx c_h$ for $b_h \leq i \leq e_h$
- Error in the description:

$$\sum_{i=1}^{n} (a_i - c_{r(i)})^2 = \sum_{j=1}^{k} \sum_{i=b_j}^{e_j} (a_i - c_j)^2$$

- Why this (L_2^2) metric? Any metric can be used.
- (This has a probability interpretation.)

Discovered segment structure of the sequence



Some notation

- $S = (a_1, a_2, \ldots, a_n)$
- Write $S = S_1, S_2, \ldots, S_k$
- Here $S_j = (a_{b_j}, a_{b_j+1}, \ldots, a_{e_j})$
- b_j: start of segment $j = 1, \ldots, k$; $b_1 = 1$
- e_j: end of segment j, i.e., $e_j = b_{j+1} - 1$ for $j < k$, and $e_k = n$
- For $i = 1, \ldots, n$ let $r(i)$ be the unique h such that $b_h \leq i \leq e_h$

The best constant for a segment, L_p metric

???

2. Finding the best segmentation

- Some simple observations
- Dynamic programming: the principle
- The basic algorithm
- Finding the boundary points
- Complexity of the algorithm
- Examples

The best constant for a segment, L_2^2 metric

- Look at segment j
- Minimize $f(c) = \sum_{i=b_j}^{e_j} |a_i - c|^2$
- Derivative: $f'(c) = -2 \sum_{i=b_j}^{e_j} (a_i - c) = 0$
- $$\sum_{i=b_j}^{e_j} a_i = (e_j - b_j + 1)c$$
- c is the average of the points

The best constant for a segment, L_1 metric

- Minimize $f(c) = \sum_{i=b_j}^{e_j} |a_i - c|$
- c is the median of the points $a_{b_j}, \ldots, a_{e_j}$
- Proof?

Basic algorithm

```
for i=1:n for j=i:n
    cost(i,j,1) = Lperror(A(i:j),mean(A(i:j));
    % incremental solutions exist
end end

for h=2:k
    for i=1:n
        cost(1,i,h) = cost(1,i,h-1);
        for j=1:i-1
            if cost(1,j,h-1)+cost(j+1,i,1)<=cost(1,i,h)
                cost(1,i,h) = cost(1,j,h-1)+cost(j+1,i,1);
            end
        end
    end
end
```

Finding the best boundary points

```
% startsfrom(1,i,h): the starting point of the last segment
% of the best h piece segmentation of (1,i)
for i=1:n
    startsfrom(1,i,1) = 1;
end
for h=2:k
    for i=1:n
        cost(1,i,h) = cost(1,i,h-1);
        startsfrom(1,i,h) = -1; % don't know this yet
        for j=1:i-1
            if cost(1,j,h-1)+cost(j+1,i,1)<=cost(1,i,h)
                cost(1,i,h) = cost(1,j,h-1)+cost(j+1,i,1);
                startsfrom(1,i,h) = j+1;
            end
        end
    end
end
```

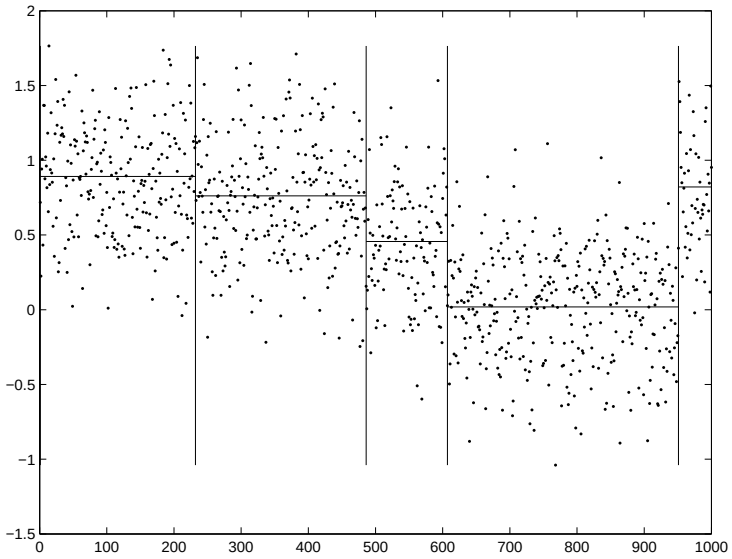
Basics

- If the segment boundaries are known, then the constants can be computed (in L_1 and L_2 metrics)
- How to find the segment boundaries?
- Exhaustive search is not feasible: too many potential segmentations
- What is the number of segmentations of n points into k segments?
- At least $\binom{n}{k-1} \approx n^{k-1}/(k-1)!$

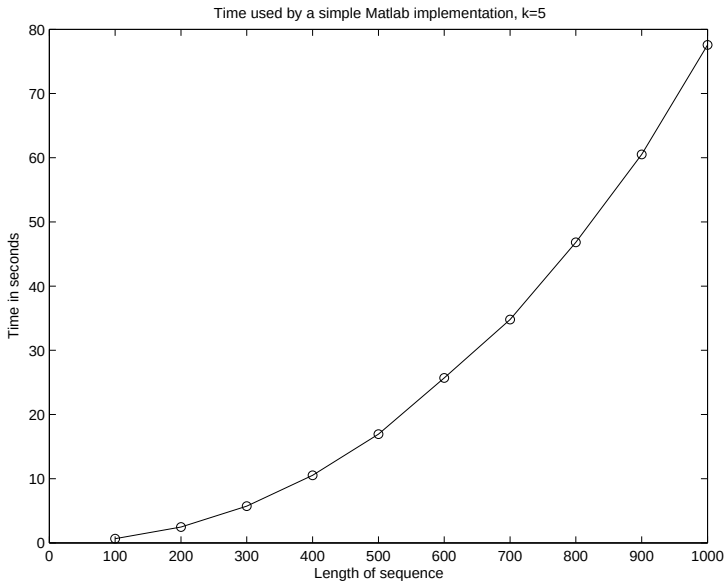
Dynamic programming principle

- The restriction of an optimal solution of an instance is an optimal solution to the restriction of the instance
- Assume $b_1, b_2, \ldots, b_k$ are the best starting points of k segments for sequence $(a_1, \ldots, a_n)$
- Then $b_1, b_2, \ldots, b_{k-1}$ are the best starting points for $(a_1, \ldots, a_{e_{k-1}})$,

Example data: 1000 points, 5 segments, normally distributed error



Example: $n = 100 \cdots 1000$, $k = 5$



Finding boundaries

```
ind = n+1; boundaries(k+1) = n; h=k;

while ind > 0 && h>0
    prevind = ind;
    ind = startsfrom(1,ind-1,h);
    boundaries(h) = ind;
    levels(h) = representative(A(ind:prevind-1),p);
    k = k-1;
end
```

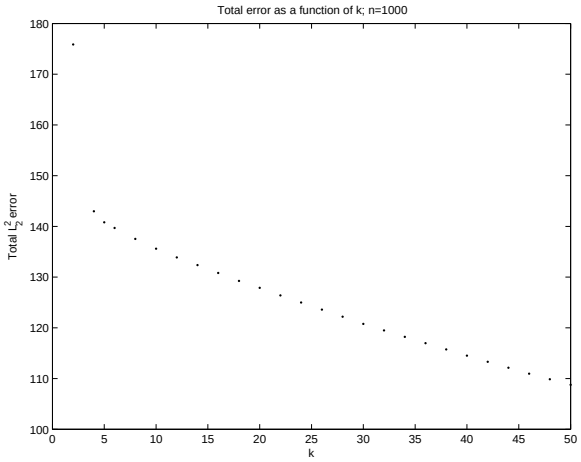
Finding the levels

- Best constant for a segment for L_2 metric is the mean, for L_1 the median.

Complexity of the algorithm

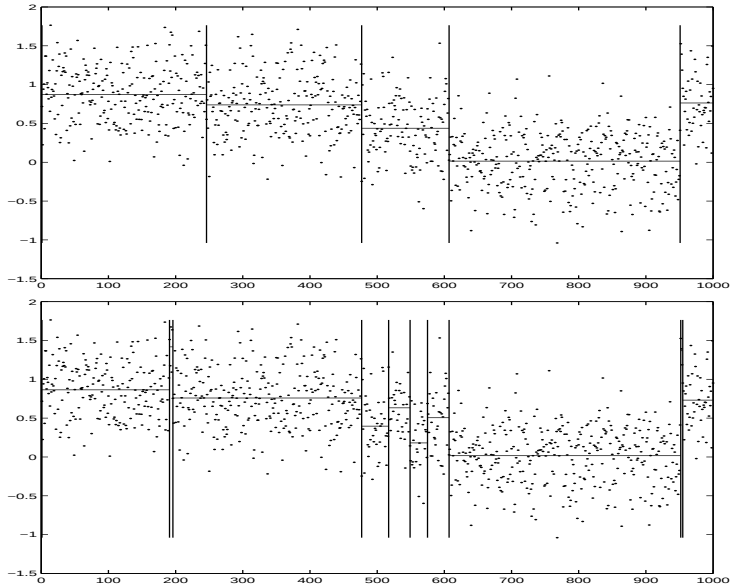
- $k = 1$: Computing the L_2 error for each segment $a_i, \ldots, a_j$
- Can be done in $O(n^2)$ time.
- $k > 1$: For each i and $j < i$ look at the solution having (j, i) as the last segment
- $O(n^2)$ time
- Total time $O(n^2 k)$.
- Is this feasible?

Dependence of error on the value of k

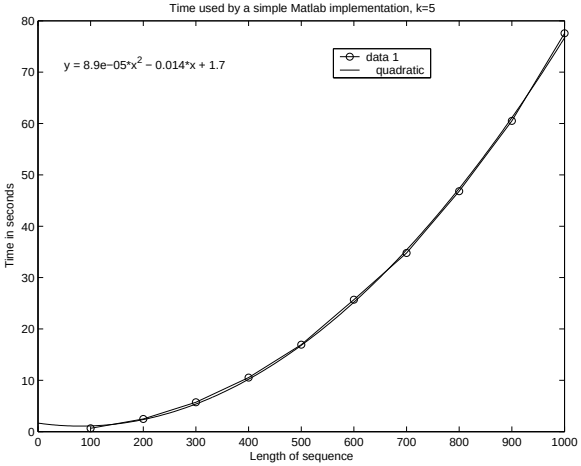


What is the correct value of k?
Will consider this later.

Comparing $k = 5$ and $k = 10$

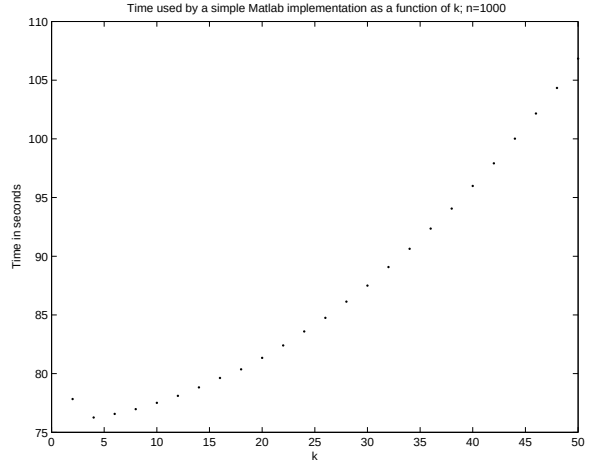


Quadratic?



Interesting, as the implementation is in principle cubic, as the $k = 1$ case is implemented in a trivial fashion.

Dependence of time on k



What types of functions can be fitted?

- Function class $PW_k(\mathcal{P})$, piecewise $\mathcal{P}$
- functions $f : \{1, \ldots, n\} \to R$ defined by k boundaries b_j, e_j and k functions $g_j \in \mathcal{P}$:
- $f(i) = g_j(i)$, where j is such that $i \in (b_j, e_j)$
- Optimization task: given $a_1, \ldots, a_n$, find $f \in PW_k(\mathcal{P})$ minimizing $\sum_i |a_i - f(i)|^2$
- The error criterion can be something else
- Algorithm: as before: compute the costs for all segments $a_i, \ldots, a_j$ and use dynamic programming to find the best collection of segments

Running time for the algorithm

- $T(m)$: the time needed to find the best function $f_j \in \mathcal{P}$ for a segment of length j
- Computing initial costs:

$$\sum_{i=1}^{n} \sum_{i'=i+1}^{n} T(i' - i + 1) = O(n^2 T(n))$$

- Dynamic programming: $O(n^2 k)$
- If the best function from class $\mathcal{P}$ for a single segment can be found in polynomial time, then the best function in the class $PW_k(\mathcal{P})$ can be found in polynomial time.

3. Generalizations

- What if piecewise constant approximation is not the appropriate way?
- In some cases piecewise linear might be better
- How to incorporate this into the basic algorithm?
- Actually quite straightforward

Example: piecewise linear

```
for i=1:n for j=i:n
    cost(i,j,1) = Lperror(A(i:j),mean(A(i:j));
end end
```

Replace by

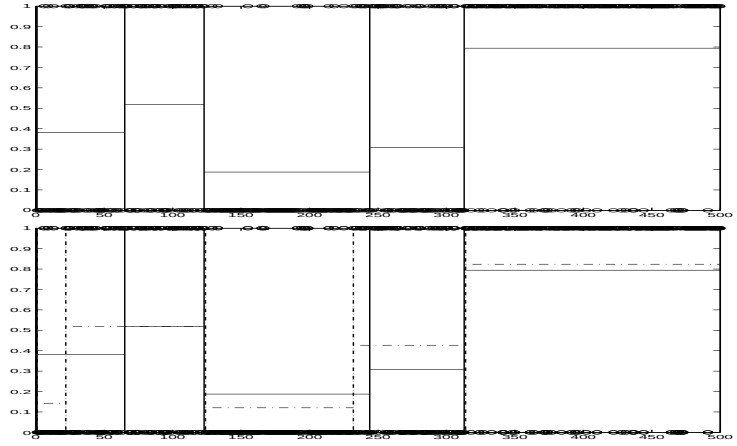
```
for i=1:n for j=i:n
    cost(i,j,1) = Lperror(A(i:j),bestlinear(A(i:j));
end end
```

Here bestlinear(A(i:j)) is the best linear approximation for the points $a_i, \ldots, a_j$, computable using linear regression.

Finding the best parameter p

- Which p maximizes $p^{n_1}(1-p)^{n_0}$?
- Derivative
  $n_1 p^{n_1-1}(1-p)^{n_0} + p^{n_1} n_0 (1-p)^{n_0-1} = 0$
- $p^{n_1-1}(1-p)^{n_0-1}(n_1(1-p) + n_0 p) = 0$
- $n_1(1-p) = n_0 p$, i.e., $(n_0+n_1)p = n_1$, i.e., $p = n_1/n$
- The natural result

Piecewise constant probabilities



levels: 0.3811 0.5180 0.1875 0.3087 0.7942

boundary points:

65 123 244 313

1 22 124 232 314 500

Probabilistic modeling of 0-1 sequences

- Somebody throws a biased coin and reports the head and tails (0s and 1s)
- Now and then the coin is changed
- How to find the changepoints?
- Data $a_1, a_2, \ldots, a_n$: sequence of 0s and 1s
- $\mathcal{P}$: probability of 1 $= p \in [0,1]$
- Error criterion: likelihood of the sequence
- Try to maximize this

Likelihood of a sequence

- $L(a_1, a_2, \ldots, a_n|p) = \prod_{i=1}^{n} L(a_i|p)$
- $L(a_i|p) = p$, if $a_i = 1$, and $L(a_i|p) = 1-p$, if $a_i = 0$
- $L(a_1, a_2, \ldots, a_n|p) = \prod_{i=1}^{n} p^{a_i}(1-p)^{1-a_i} = p^{n_1}(1-p)^{n_0}$
- $n_1 =$ number of 1s in the sequence
- $n_0 =$ number of 0s in the sequence $= n - n_1$

Does it matter whether one optimizes loglikelihood or squared error?

- $\sum_i |a_i - p|^2$ vs.
  $\sum_i (a_i \log p + (1 - a_i) \log(1 - p))$
- $n_1(1-p)^2 + n_0 p^2$ vs. $n_1 logp + n_0 \log(1-p)$
- The optimal p for a $\mathbf{single}$ segment is the same
- What about the optimal segmentations?
- Not always the same