Recent advances in multiple change-point detection

Piotr Fryzlewicz

London School of Economics, UK

Vienna University of Economics and Business, June 2017

Introduction – nonparametric estimators as algorithms

Estimators formulated as optimizers of a certain criterion have always featured in the statistical literature and continue to do so. As their complexity increases, so does the numerical effort needed to compute them.

Obtaining their values often requires non-trivial expertise (e.g. can I compute the Lasso estimator if I do not know linear programming?) and can sometimes go disastrously wrong.

By contrast, our preference is for estimators directly formulated as algorithms which already incorporate the computation part.

Introduction – nonparametric estimators as algorithms

Formulating estimators as algorithms can make our life easier in at least two ways:

- Unlike in traditional "write as optimum + compute" estimators, algorithmically formulated estimators can give a better idea of the "real" estimation error, as there is no extra error hidden in the computation part.
- Occasionally, formulating estimators as algorithms can in fact help in their theoretical analysis, as this talk will try to demonstrate.

Greedy algorithms

This talk will be about two particular classes of greedy algorithms.

Greedy procedures can be an attractive alternative to full likelihood-based methods, when the latter are slow to compute or the precision of computation is difficult to control.

There is no need to view greedy methods as "approximating" the corresponding full-likelihood ones: they are often simple enough for their properties to be tractable in their own right, and can be (near-)optimal.

Part I: Wild Binary Segmentation and Narrowest-Over-Threshold detection of multiple change-points

Part I

Wild Binary Segmentation and Narrowest-Over-Threshold detection of multiple change-points

(NOT part joint with Rafal Baranowski, now at Winton Capital Management, and Yining Chen, LSE)

Problem set-up

Consider the model

$$X_t = f_t + \varepsilon_t,$$

where the unobserved function f_t contains an unknown number of point "features" at unknown locations, and ε_t is centred noise.

Examples:

- change-point detection (f_t piecewise-constant),
- knot selection in spline smoothing,
- trend changes in time series analysis.

A "feature" can be anything we know how to estimate the location of, if we know that there is only one present.

Motivating example: change-point detection 1

The simplest a posteriori change-point detection problem is when f_t is piecewise-constant, and we need to estimate the number and locations of the jumps in f_t.

Two main classes of solutions in the literature:

1. Those that estimate all change-points at once, often via a likelihood-type fit of a piecewise constant function to data X_t, plus penalty for the number (often) or locations (less often) of change-points.

   Pros: accurate in theory, weak theoretical assumptions.
   Cons: estimators can be slow to compute; a lot depends on the quality of the optimizer involved, whose behaviour is often poorly understood. Often unclear what penalty is best (if any).

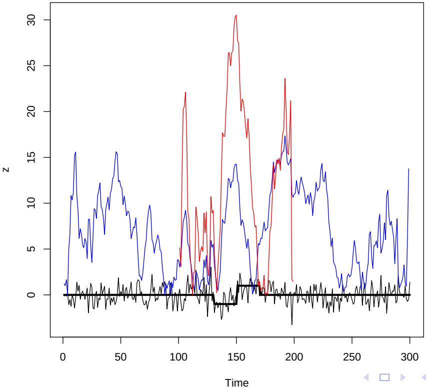
Motivating example: change-point detection 2

2. Those that estimate the change-points one by one. Well-known example: Binary Segmentation, in which change-points are estimated greedily one by one and the current sample is iteratively split into two by each estimated change-point.

   Pros: fast to compute, conceptually simple, easy to code. **Cons:** good theoretical performance requires stronger theoretical assumptions, convergence rates poor in more challenging settings.

Improving Binary Segmentation: "Wild" Bin Segmentation

The following example illustrates potential issues with standard Binary Segmentation. Data X_t in black, global CUSUM in blue, local CUSUM in red (CUSUM is a least-squares measure of the quality of the fit of a piecewise-constant function with one jump to the data):



Wild Binary Segmentation

Clearly, it would have been preferable to use the maximum of the red curve as a locator for a change-point candidate. However, it is obviously not clear a priori what starting point s and end-point e to choose.

Motivated by this, in our earlier work we proposed the following **Wild Binary Segmentation** (WBS) locator statistics

$$WBS = \arg\max_{s^*, b, e^*} |\mathrm{CUSUM}_{s^*, b, e^*}(X)|,$$

where s^*, e^* are drawn uniformly over the current data segment $[s, e]$ a suitable number of times. Checking all s^*, e^* would have resulted in cubic computational complexity, which would be prohibitive – hence the random draws.

The b that achieves the above maximum is taken as a change-point candidate. The procedure then continues in the usual binary way.

Wild Binary Search for other features?

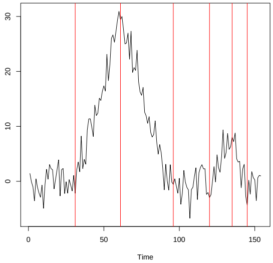
Question: can a similar principle be applied in other nonparametric problems?

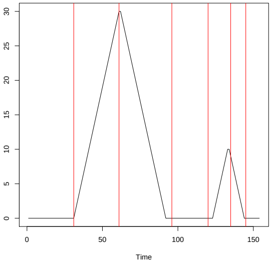
Postulated generic algorithm:

1. Extract a large enough number of (randomly, or deterministically specified) sub-segments of the data, covering the entire domain well enough.
2. Fit the "best" simple feature on each sub-segment.
3. Compare the fits over the drawn sub-segments and choose the best fit overall.
4. If this is found to be significant, record it, remove its effect, and go to step 1.

Wild Binary Search for other features? – contd

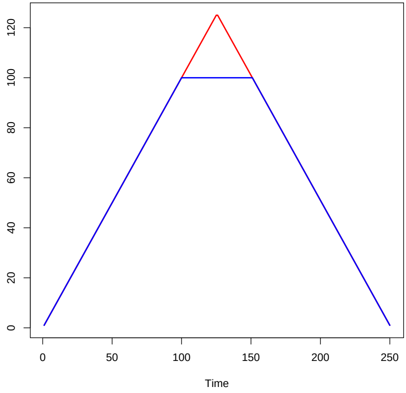
Another "successful" case? "Wild" detection of kinks (discontinuities of the first derivative), using a triangular-shaped contrast function. 6 most important detected features marked in red.





Wild Binary Search for other features? – contd

Not so fast! This, in fact, does *not* work. The reason is simple: in this case, if the current interval contains two or more features, it may happen that the best approximation by one feature will not indicate either of them:



Wild Binary Search for other features – our proposal

We circumvent this issue via the following trick. Instead of the contrast with the largest value

$$\arg \max_{s^*,b,e^*} |\mathrm{Contrast}_{s^*,b,e^*}(X)|,$$

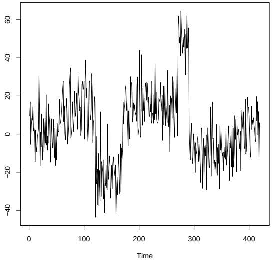
we prefer the "narrowest" one that exceeds a certain threshold λ:

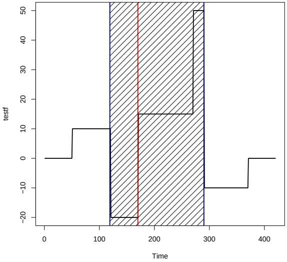
$$\arg \min_{s^*, \arg\max_b |\mathrm{Contrast}_{s^*,b,e^*}(X)|, e^*} \{|e^* - s^*| : \max_b |\mathrm{Contrast}_{s^*,b,e^*}(X)| > \lambda\}.$$

We term this criterion "Narrowest-Over-Threshold" (NOT).

NOT criterion: example

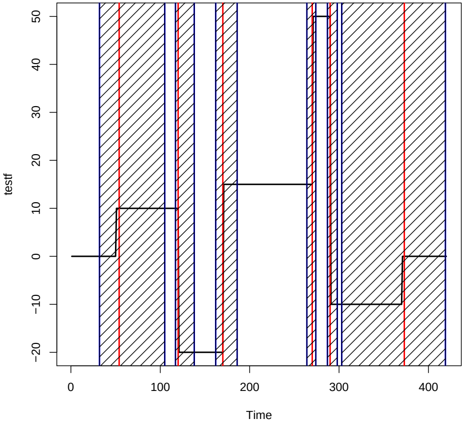
The following example is in the context of change-point detection. Consider the simulated signal on the left. The first interval returned by the classical WBS criterion together with the corresponding change-point estimate on the right. The interval straddles at least 2 features.





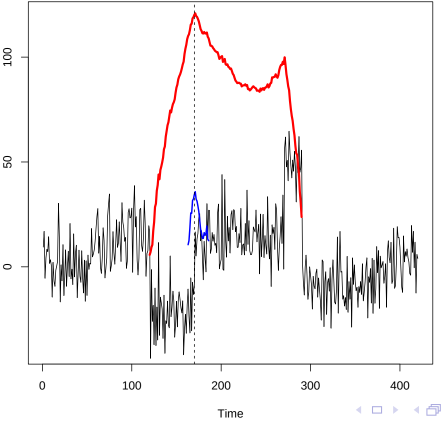
NOT criterion: example contd

. . . and here are the features detected by the NOT criterion:



NOT criterion: example contd

Comparison of the CUSUM shapes used to detect the change-point at $t = 170$ used by WBS (red) and NOT (blue). In NOT, it is not the absolute magnitude of the CUSUM that matters, but the "sharpness".



NOT criterion: theory and comments

1. It is no coincidence that each interval contains exactly one feature: this can be ensured to happen in theory (with probability tending to 1 with T), which makes it straightforward to prove the consistency of NOT in multiple feature detection as long as we know how to detect a single feature.
2. This completely overcomes the "two feature" problem illustrated earlier and hence enables the use of many different detection statistics.
3. The initial sorting includes the extra threshold parameter λ, but the final model can still be chosen in a "threshold-less" way if desired (e.g. via an information criterion).
4. Advantage over approaches such as "moving CUSUM" as no need to choose a bandwidth parameter.
5. Competitors: adaptive splines (Mammen and van de Geer), trend filtering (Tibshirani).

NOT paper: summary and example results

Heuristic summary of theoretical results from the 'NOT' paper (references at end of talk):

- In the piecewise-constant and piecewise-linear models with N change-points, $\hat{N} = N$ on a set of large probability, and locations estimated with near-optimal rates.
- 'Roadmap' on how to prove consistency for higher-order piecewise-polynomial settings.
- Important aspect: fast computation of the entire solution path.

Part II: Tail-greedy algorithms

Part II

Fast ‘tail-greedy’ bottom-up decomposition algorithms for signals on graphs and network adjacency matrices

(network part joint with Xinyu Kang and Eric Kolaczyk, Boston U)

Introduction – "object-oriented" thinking in data analytics

This talk advocates object-oriented thinking in data analytics. By this we mean designing methods and code so that they work similarly for a range of data structures, or at least exhibit strong similarities / transferable elements.

A popular mantra at statistics meetings these days is that modern methods need to find a balance between statistical accuracy and computational costs. Some are even explicitly formulated to achieve this.

One often forgotten component of this trade-off is the time-person cost of writing computer code. Designing methods in an "object-oriented" way can go some way towards reducing this cost.

Motivating example – change-point detection

We start the description of our methodology on the simplest example – one-dimensional data in the signal+noise model:

$$X_t = f_t + \varepsilon_t, \quad t = 1, \dots, T,$$

where f_t is piecewise-constant, and ε_t are iid standard normal. The task is to detect the number N and locations η_i of change-points in f_t.

Literature on change-point detection

[Recap:] Two main types of approaches in the literature:

1. **Optimise (fit to data + penalty on number of changepoints).** Can be rate-optimal in theory, but can be slow in practice, requires advanced coding skills, penalty choice not very intuitive, which may lead to inaccurate results in practice.
2. **Binary segmentation.** Detect change-points one by one in a greedy divisive way, from the top down. Fast and simple to code, but can be rate-suboptimal in theory and often does not perform well in more challenging situations.

Bottom-up alternatives?

Another problem (from the “object-oriented” perspective) is that neither of the two methodologies (not even improved versions of 2, such as Circular Binary Segmentation or Wild Binary Segmentation) extend naturally to other change-point-like problems such as edge detection in images, or community detection in networks.

In this talk, we propose a new methodology that

- is fast,
- is an agglomerative, bottom-up alternative to the divisive, top-down binary segmentation,
- introduces the concept of ‘tail-greediness’, which induces both fast speed and statistical consistency,
- has some near-optimality properties,
- has a natural extension to other change-point-like problems such as those described above.

Outline of the procedure on an illustrative example

Outline of the procedure.

The decomposition part. Input: vector $X_1, \ldots, X_T$. (In this example, $T = 9$).

1. Current vector := input, $n := T$.
2. Applying suitable filters (a_i, b_i) s.t. $a_i^2 + b_i^2 = 1$, compute inner products between those filters and all pairs of neighbours in the current vector (= "details"). Filters to be such that if the input vector is constant over their support, the inner products are zero. Fuse those components of the current vector for which these (absolute) inner products are among the $\lceil \rho n \rceil$ smallest, taking care that they do not overlap. Record the corresponding details, and assign new data points to the fused regions computed using filters orthonormal to those used to compute details.
3. $n :=$ length of the new current vector. Go back to step 2. Continue as long as possible.

Example of how the procedure may progress

Example:

1st pass: $\left(X_1, X_2, \frac{X_3+X_4}{\sqrt{2}}, X_5, \frac{X_6+X_7}{\sqrt{2}}, X_8, X_9\right)$. Record $\frac{X_3-X_4}{\sqrt{2}}, \frac{X_6-X_7}{\sqrt{2}}$ (two smallest "details" out of those examined).

2nd pass: $\left(X_1, X_2, \frac{\sqrt{2}}{\sqrt{3}}\left(\frac{X_3+X_4}{\sqrt{2}}\right) + \frac{1}{\sqrt{3}}X_5, \frac{X_6+X_7}{\sqrt{2}}, \frac{X_8+X_9}{\sqrt{2}}\right)$. Record $\frac{1}{\sqrt{3}}\left(\frac{X_3+X_4}{\sqrt{2}}\right) - \frac{\sqrt{2}}{\sqrt{3}}X_5, \frac{X_8-X_9}{\sqrt{2}}$ (two smallest "details" out of those examined).

Continue in a similar fashion for as long as possible. The number of passess will be $O(\log T)$.

Some comments

Some comments on the procedure.

1. The resulting transformation is non-linear, but conditioning on the order in which the details are selected, it is orthonormal.
2. Computational complexity is: $O(\log\ T)$ passes through data $\times$ $O(T \log\ T)$ operations in each pass.
3. The term tail-greedy comes from the fact that in each pass, details in the lower tail of their distributions get selected. Tail-greedy is 'less greedy than greedy'. Tail-greediness causes the method to terminate in at most $O(\log\ T)$ passes.
4. Since the transform is conditionally orthonormal, it preserves the l_2 norm of the input signal. Because the initial fine-scale coefficients are forced to be small, most energy will be concentrated at coarser scales.
5. Natural alternative to "fused lasso"-type approaches.
6. Extends easily to images and other signals on graphs.

Signal estimation and change-point detection

Estimating f:

1. Take the transform
2. Threshold away "small" detail coefficients. Smallest admissible threshold is smaller than $\lambda = 2\{(1+\delta)\log T\}^{1/2}$, δ arbitrary positive. Preserve the connectedness of the tree of coefficients, i.e. only threshold away a coefficient if it and all its "children" fall under the threshold.
3. Take the inverse transform.

As a result, with probability tending to 1,

$$T^{-1}\|\hat{f}-f\|^2 \leq 2(1+\delta)\,T^{-1}\log T\left\{1+(3+2\sqrt{2})N\lceil\log T/\log\{(1-\rho)^{-1}\}\rceil\right\}.$$

If the algorithm were greedy, but not tail-greedy, consistency would not hold.

Simple post-processing also yields consistency in change-point detection.

Examples

Speed of execution (written in “pure” R in the sense that contains no own non-R code):

```
> system.time(segment.mean(rnorm(10^5)))
  user   system elapsed
  4.06    0.26    4.32
```

The following examples are the same as in

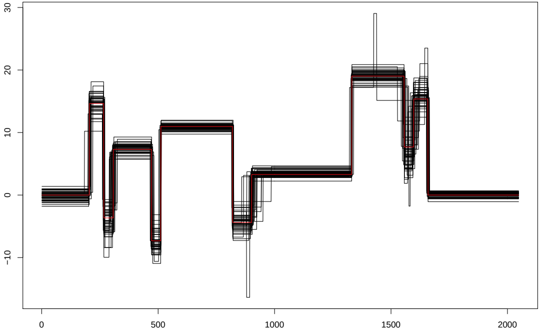
Wild Binary Segmentation for multiple change-point detection. P. Fryzlewicz (2014). Ann. Stat., 42, 2243-2281.

Examples contd – comparative performance

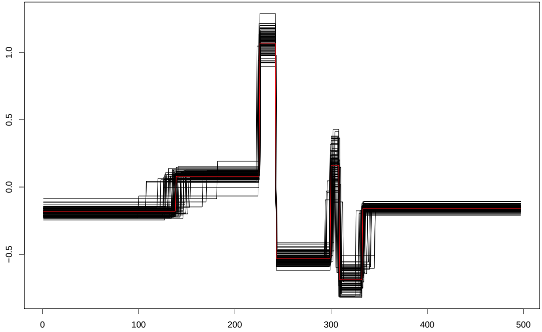
	TGUH	PELT	S3IB	WBS
blocks	44	41	45	46
fms	84	91	86	95
mix	38	12	12	33
teeth10	68	36	48	80
stairs10	92	95	0	61
average	65.2	55.6	38.2	63

Table: The number of times, out of 100 simulated sample paths, that $\hat{N} = N$ was achieved for the various competing methods and models, along with averages over the models considered.

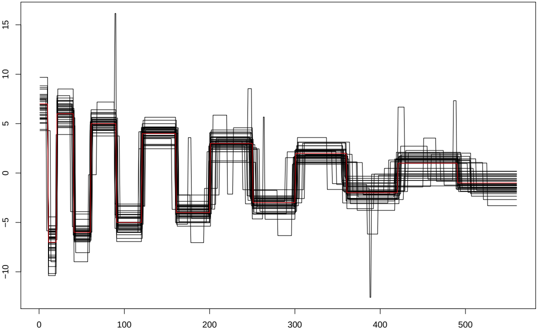
Examples contd – the “blocks” signal



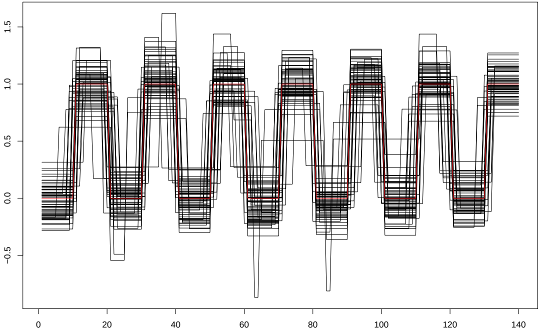
Examples contd – the “fms” signal



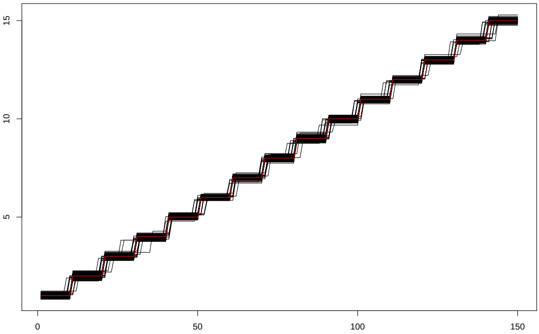
Examples contd – the "mix" signal



Examples contd – the “teeth10” signal



Examples contd – the “stairs10” signal



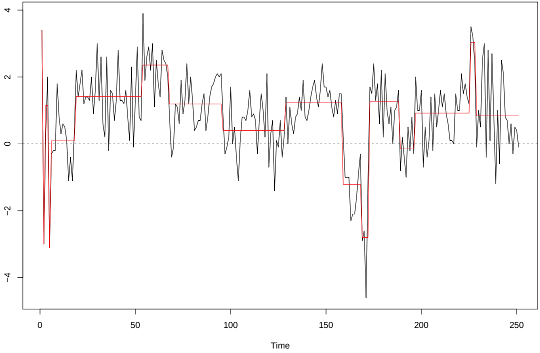
Data example – London house prices

We analyse monthly percentage changes in the UK's Land Registry House Price Index (HPI), from January 1995 to December 2015, in three east London boroughs: Hackney, Newham and Tower Hamlets.

Data available from `http://landregistry.data.gov.uk/app/hpi`.

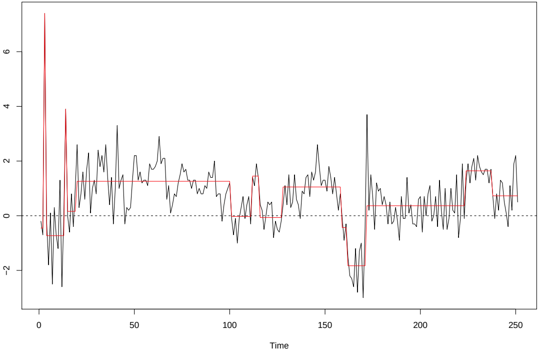
Data example – London house prices

Black: monthly percentage changes in the HPI for Hackney, from January 1995 to December 2015. Red: the TGUH estimate with default parameters.



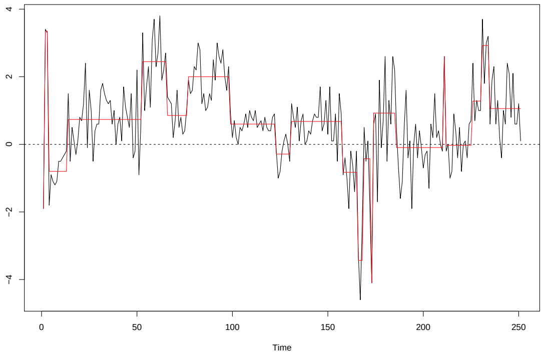
Data example – London house prices

Black: monthly percentage changes in the HPI for Tower Hamlets, from January 1995 to December 2015. Red: the TGUH estimate with default parameters.



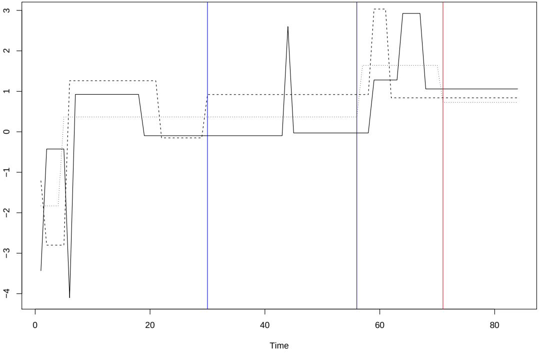
Data example – London house prices

Black: monthly percentage changes in the HPI for Newham, from January 1995 to December 2015. Red: the TGUH estimate with default parameters.



Data example – London house prices

TGUH estimates for Newham (solid), Hackney (dashed) and Tower Hamlets (dotted), from January 2009 to December 2015. Blue lines: June 2011 and August 2013; red line: November 2014.



Data example – London house prices

For extended periods of time, HPI increases in Newham trailed those in the other two boroughs, despite the investment in Newham related to the 2012 Olympics. In particular, this is clearly seen between June 2011 and August 2013, when (except the strong positive spike in August 2012), the average HPI increase in Newham is negative. One commentary in the Guardian, a major UK newspaper, attributes this to what it sees as an over-supply of new properties in Newham.

Data example – London house prices

Interestingly, the situation is reversed in the more recent time period November 2014 to December 2015, in which Newham shows the strongest increases in the HPI among the three boroughs. Some newspapers and online news sources speculate that this may have been due not only to the regeneration taking place in the borough of Newham, but also to some buyers having been priced out of the more central (and expensive) boroughs of Hackney and Tower Hamlets.

Extension to network decompositions

(Joint work with Xinyu Kang and Eric Kolaczyk, Boston U.)

Let us see on an example how the same idea can be applied to obtaining 'tail-greedy', bottom-up decompositions of networks. We will act on their adjacency matrices. Consider the following example.

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    0    0    1    1
[2,]    1    1    0    1    0    0
[3,]    0    0    1    0    0    0
[4,]    0    1    0    1    0    1
[5,]    1    0    0    0    1    1
[6,]    1    0    0    1    1    1
```

Outline of the procedure

Here is how we proceed.

1. Search for pairs of columns corresponding to pairs of linked nodes for which the “detail” coefficients (defined as before but where X_i are now entire columns of the adjacency matrix) are the smallest (in a certain pre-chosen norm).
2. Record the details, fuse the columns as before.
3. Symmetrise the adjacency matrix by performing the analogous operations on the corresponding rows.
4. Go to step 1.

Example contd

Suppose we only wish to fuse one pair of columns in the first pass (in reality it will be proportion ρ). The closest columns in the L_1 norm are 5 and 6 (1 and 5 too, but we cannot fuse in the same pass two different pairs containing the same node).

Below is the result of:

1. replacing column 5 with fused old columns 5 and 6;
2. replacing column 6 with detail between old columns 5 and 6.

```
     [,1] [,2] [,3] [,4] [,5]  [,6]
[1,]    1    1    0    0 1.41  0.00
[2,]    1    1    0    1 0.00  0.00
[3,]    0    0    1    0 0.00  0.00
[4,]    0    1    0    1 0.71 -0.71
[5,]    1    0    0    0 1.41  0.00
[6,]    1    0    0    1 1.41  0.00
```

Example contd

Finally, we “symmetrise” the matrix by performing analogous rotation(s) of the corresponding rows.

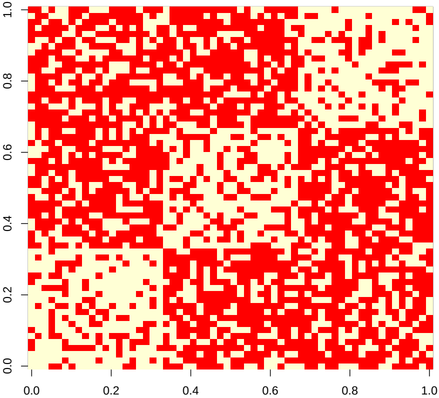
```
     [,1] [,2] [,3]  [,4] [,5]  [,6]
[1,]  1.0    1    0  0.00 1.41  0.00
[2,]  1.0    1    0  1.00 0.00  0.00
[3,]  0.0    0    1  0.00 0.00  0.00
[4,]  0.0    1    0  1.00 0.71 -0.71
[5,]  1.4    0    0  0.71 2.00  0.00
[6,]  0.0    0    0 -0.71 0.00  0.00
```

Since we have only performed rotations, this preserves the Frobenius norm of the matrix and produces a hierarchical, adaptive, multiscale decomposition of the network. In the next pass, we only act on `[1:5,1:5]`.

Network "denoising": community detection?

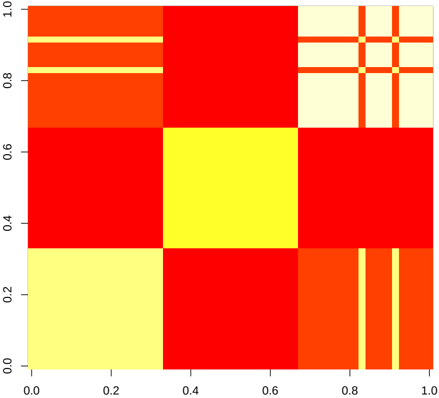
It is possible to "denoise" the adjacency matrix by thresholding away the smaller details and inverting the decomposition. This can be useful in community detection and network simulation.

Example: network sampled from the block model –



Network "denoising": community detection?

And its "denoised" version by retaining the two final detail coefficients:



References

References:

- Wild Binary Segmentation for multiple change-point detection. P. Fryzlewicz (2014). *Annals of Statistics*, 42, 2243-2281.
- Unbalanced Haar technique for nonparametric function estimation. P. Fryzlewicz (2007). *JASA*, 102, 1318–1327.
- SHAH: SHape-Adaptive Haar wavelets for image processing. P. Fryzlewicz and C. Timmermans (2016). *JCGS*, 25, 879-898.
- Tail-greedy bottom-up data decompositions and fast multiple change-point detection. P. Fryzlewicz (2016). Under revision.
- Narrowest-Over-Threshold detection of multiple change-points and change-point-like features. R. Baranowski, Y. Chen and P. Fryzlewicz (2016). In submission.
- R packages (on CRAN): **breakfast**, **not**, **unbalhaar**, **wbs**.