Data-analysis applications of the fused lasso and related spatial smoothers

James Scott

describing work with:
Oscar Padilla
Wesley Tansey
Alex Athey
Nick Polson
Alex Reinhart

Adapted from a talk at ISBA 2016

A recurring problem

A surprisingly wide array of data-analysis problems can be addressed by solving an optimization problem of the following form.

$$\underset{\beta}{\text{minimize}} \quad l(\beta) + \lambda \|D\beta\|_1 \,,$$

where β is a signal on a graph $\mathcal{G}$, $l(\beta)$ is a loss function, λ a penalty parameter, and D a discrete smoother on the graph.

A recurring problem

A surprisingly wide array of data-analysis problems can be addressed by solving an optimization problem of the following form.

$$\underset{\beta}{\text{minimize}} \quad l(\beta) + \lambda \|D\beta\|_1 \,,$$

where β is a signal on a graph $\mathcal{G}$, $l(\beta)$ is a loss function, λ a penalty parameter, and D a discrete smoother on the graph.

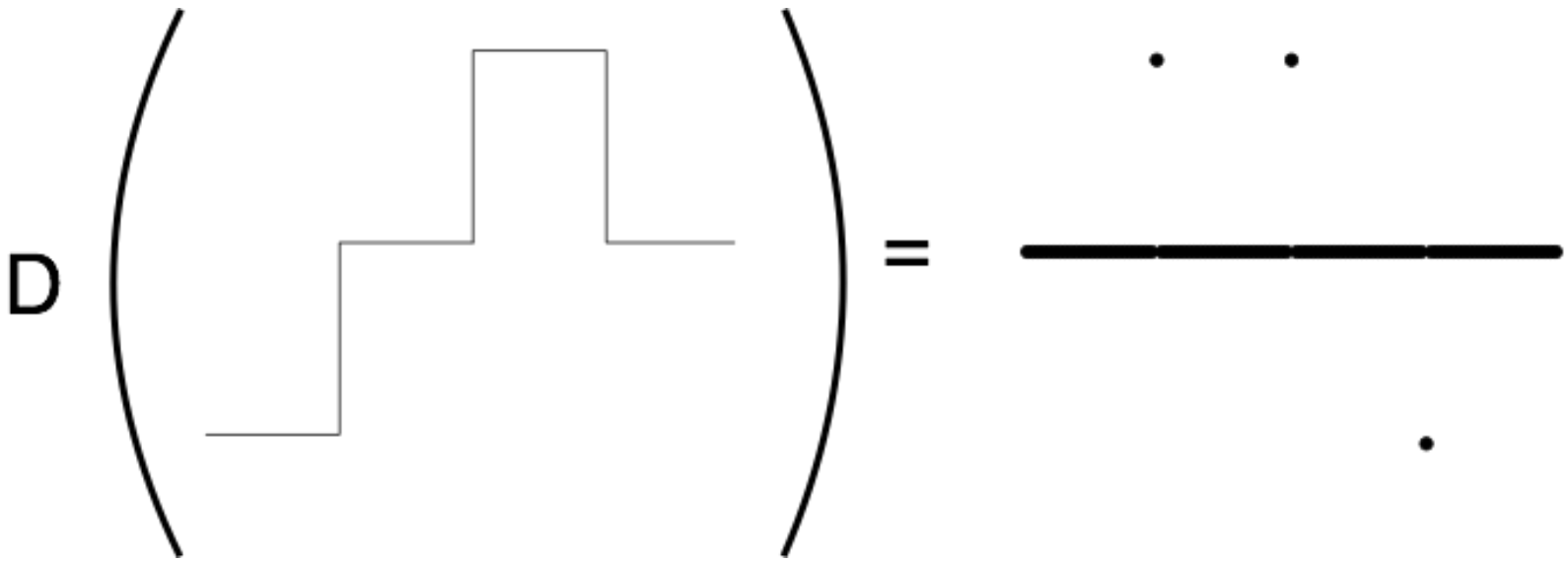
A simple case is the **1D fused lasso**. Observe noisy data $y_i = \beta_i + e_i$ at equally spaced points on the x axis. Estimate β by solving

$$\underset{\beta}{\text{minimize}} \quad \frac{1}{2}\|y - \beta\|_2^2 + \lambda \sum_{i=2}^{N} |\beta_i - \beta_{i-1}| \,.$$

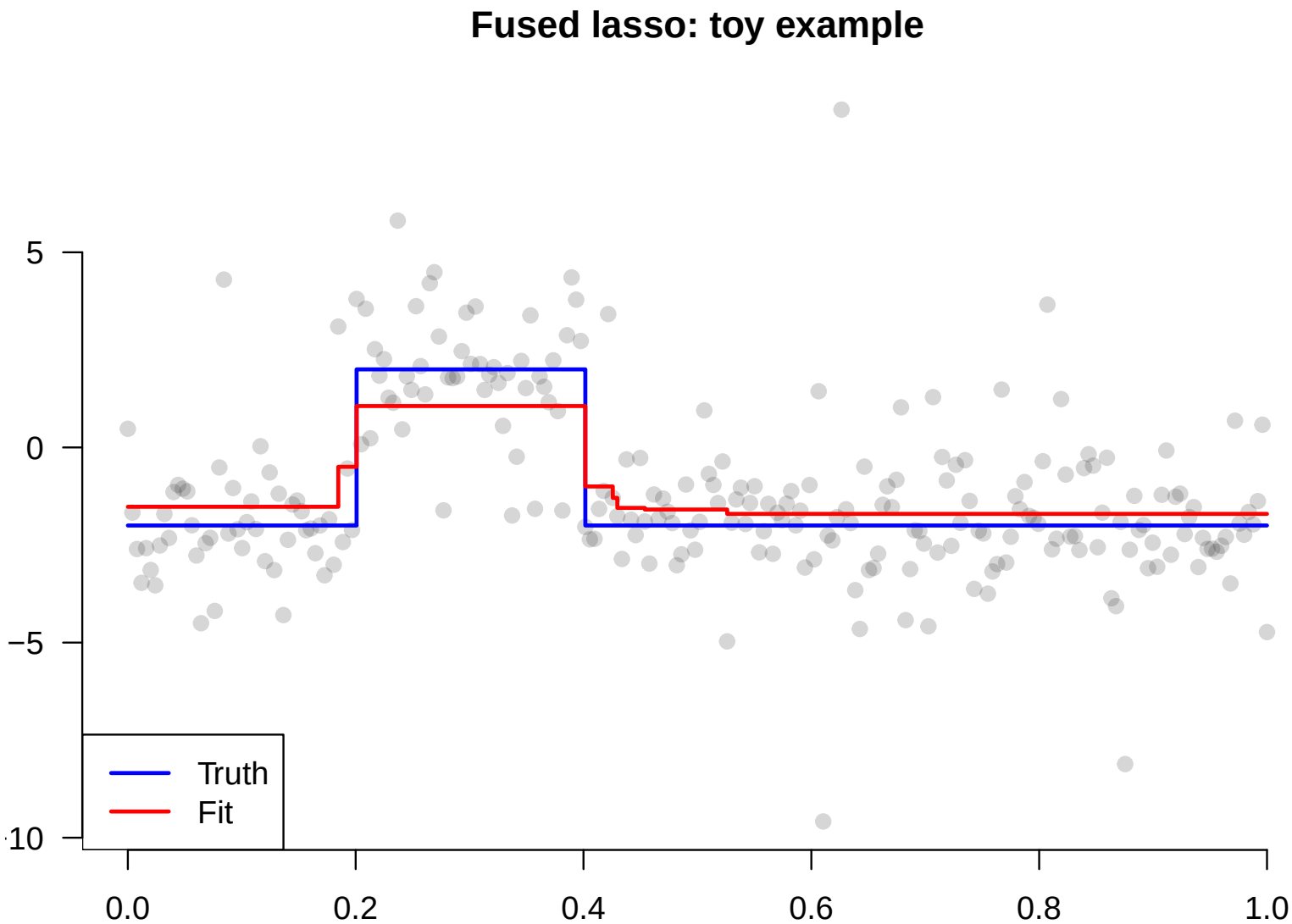
Fused lasso

Here D is the order-1 difference matrix on a chain graph:

$$\sum_{i=2}^{N} |\beta_i - \beta_{i-1}| = \|D\beta\|_1 , \quad D = \begin{pmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & \cdots & & 0 & 1 & -1 \end{pmatrix}$$



Fused lasso



Talk outline

The modern frequentist approach to spatial smoothing on discrete lattices

- ℓ_1 trend filtering (Kim et. al, 2009; Tibshirani, 2014)
- Graph fused lasso and graph trend filtering
- Lots of toy examples

How we've extended this framework in new applied directions

- Estimating a spatially varying density function
- Multiple testing
- Density estimation and deconvolution

Talk outline

Some difficulties with the classical framework

- non-diminishing bias of the ℓ_1 penalty ("sandpaper")
- speed and stability beyond the simpler cases
- choosing the penalty parameter outside squared-error loss
- no error bars

How a Bayesian approach can address some these issues*

- Bayesian graph trend filtering

*When the problem is of a moderate size.

Trend filtering

The trend-filtering estimator $\hat{\beta}$ is defined as the solution to

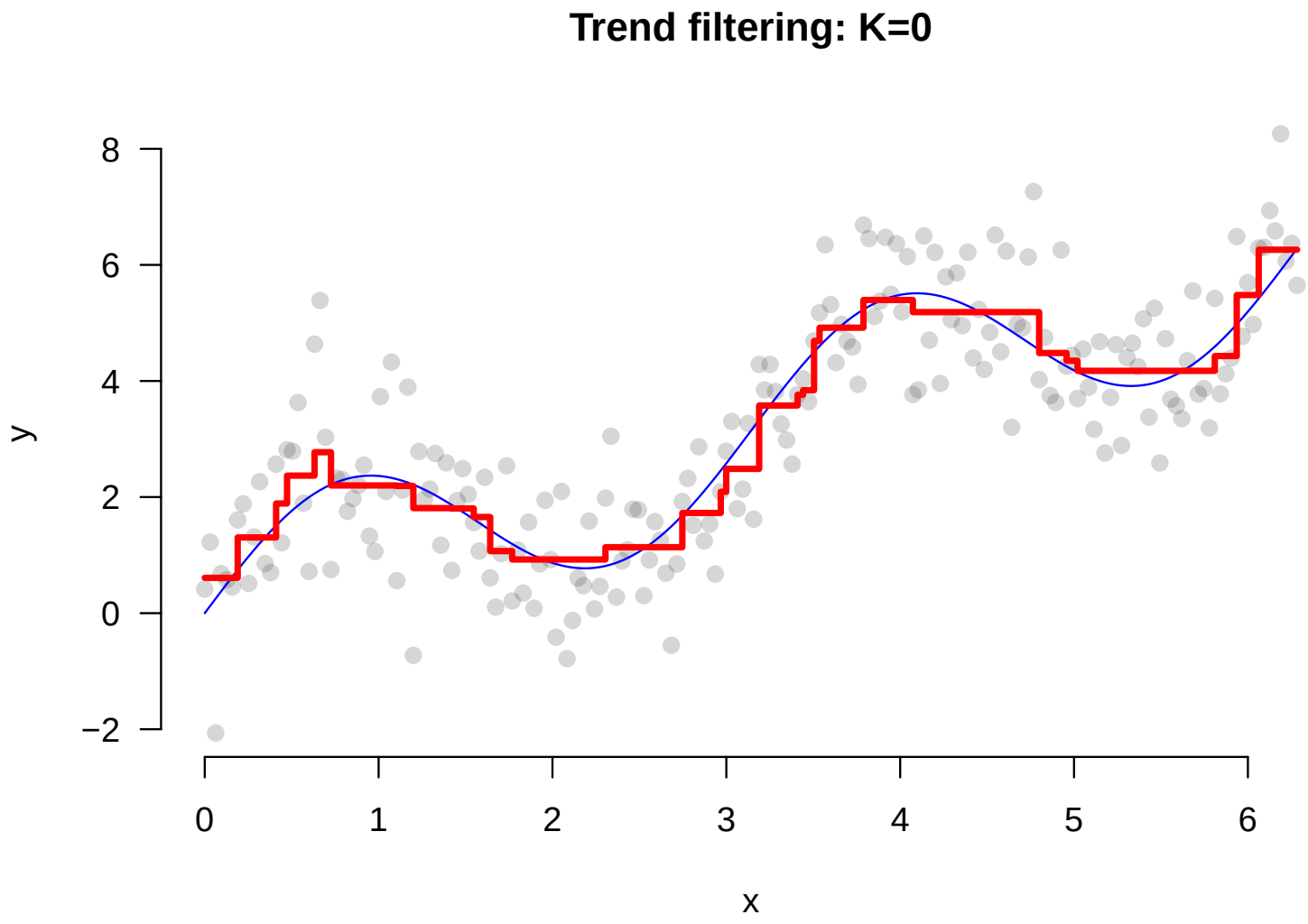
$$\underset{\beta \in \mathcal{R}^N}{\text{minimize}} \quad \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|D^{(k+1)}\beta\|_1 ,$$

where $D^{(k+1)}$ is the discrete analog of the order-k derivative. When $k = 0$, this is the fused lasso:

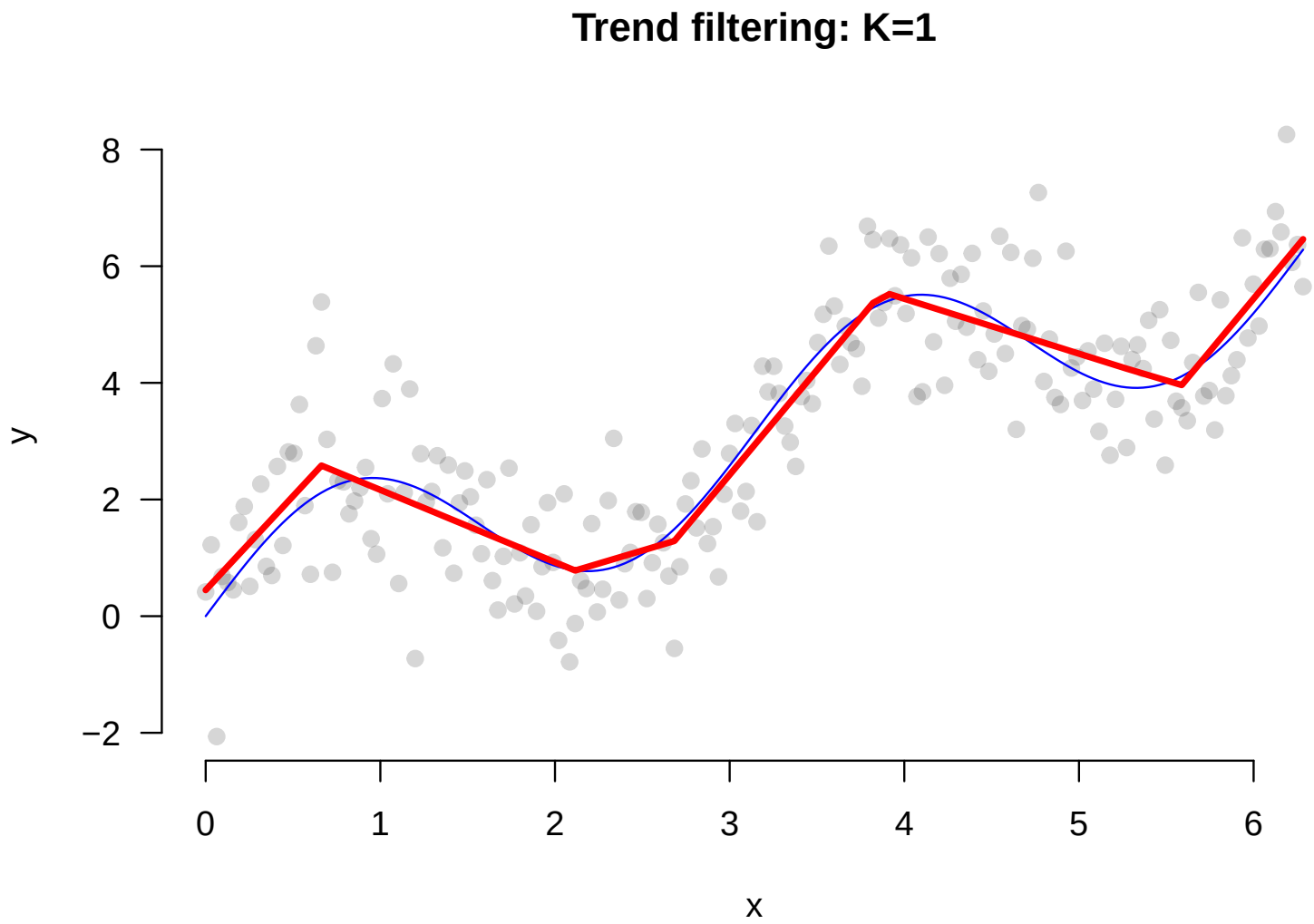
$$D^{(1)} = \begin{pmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & \cdots & & 0 & 1 & -1 \end{pmatrix}. \tag{1}$$

For $k \geq 1$ this matrix is defined recursively: $D^{(k+1)} = D^{(1)} D^{(k)}$.

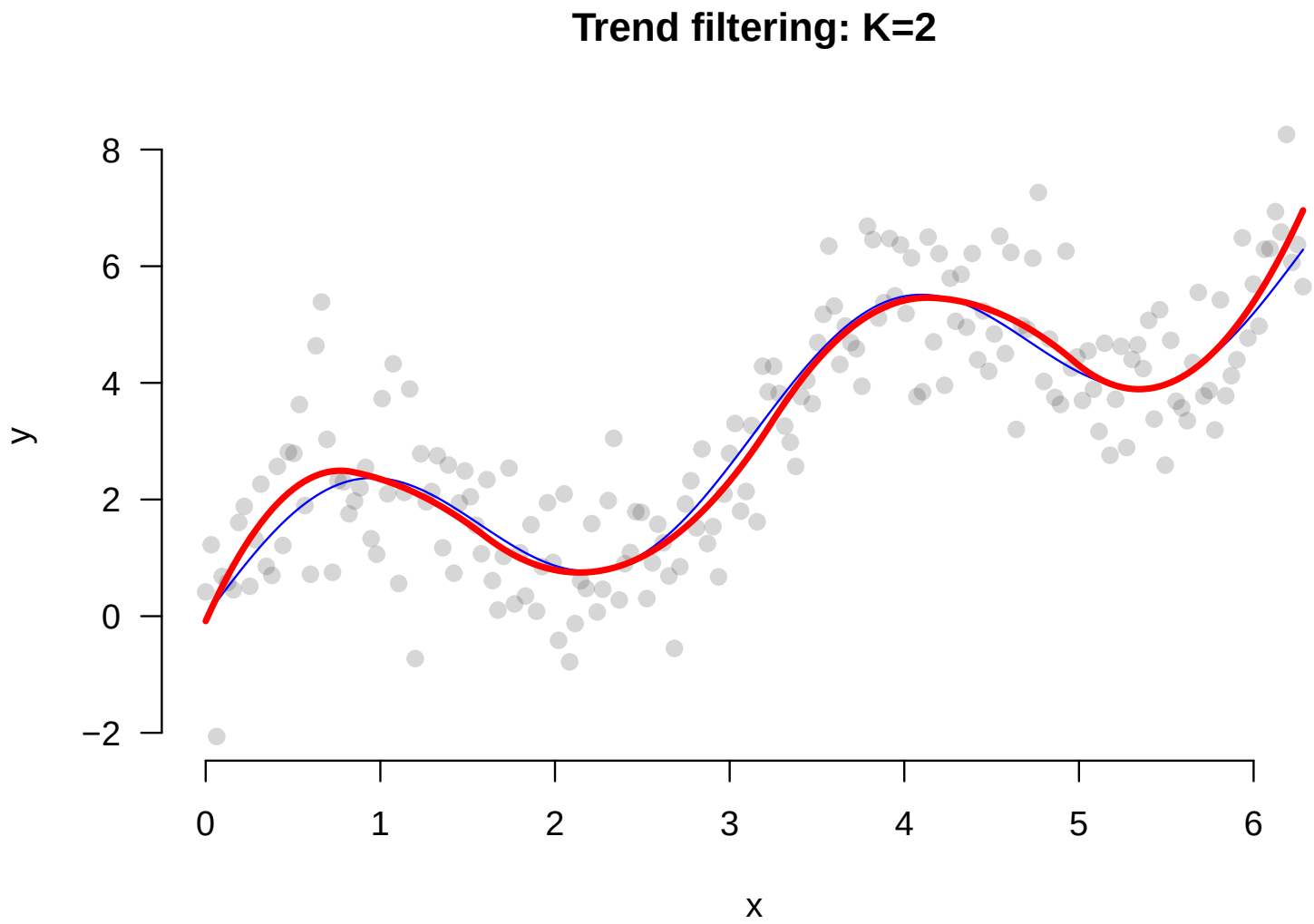
Trend filtering: toy example



Trend filtering: toy example



Trend filtering: toy example



Why are these popular?

Speed:

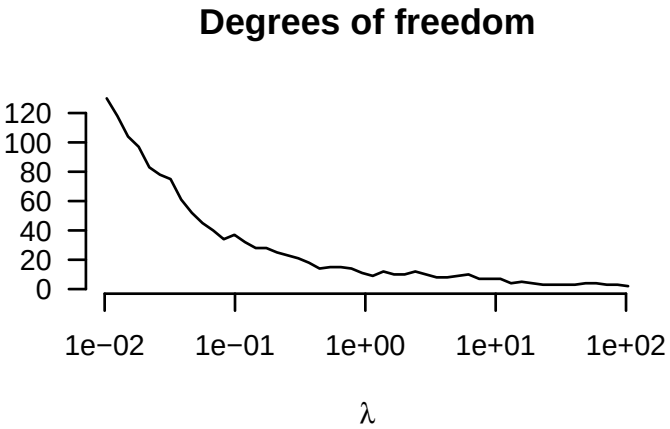
- The 1D fused lasso can be solved in $O(n)$ time.
- Trend filtering is also very fast (Ramdas and Tibshirani, 2014).

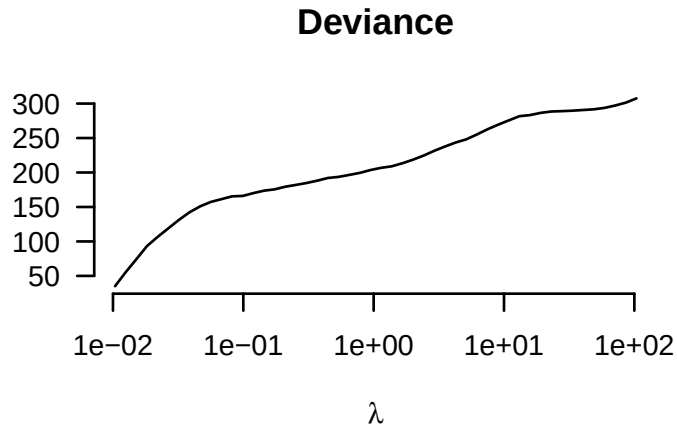
Well-known properties:

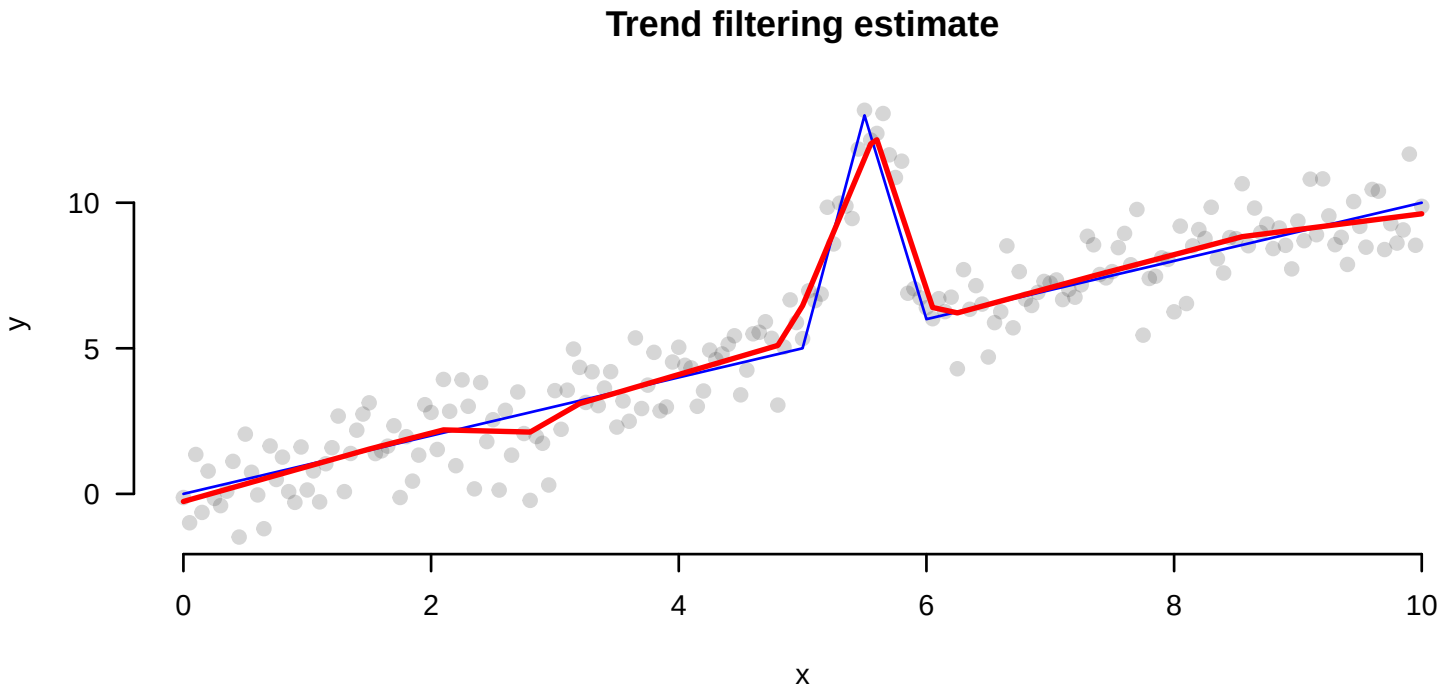
- Estimates converge at the minimax rate for functions whose order-k derivative is of bounded variation.
- Similar accuracy to locally adaptive regression splines, but much faster ($n = 10^6$: trend filtering still tractable).
- Very effective for signals with inhomogeneous spatial variation.
- $\mathrm{df}(\hat{\beta}_\lambda)$ is known (Tibshirani and Taylor, 2011). Recall for $g(y) : \mathcal{R}^n \to \mathcal{R}^n$,

$$\mathrm{df}(g) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{cov}(g_i(y), y_i)$$

Why are these popular?

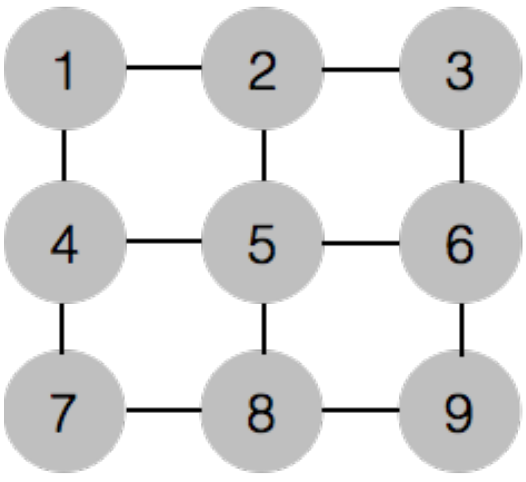






Graph fused lasso

Both procedures can be generalized beyond chain graphs. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, and let D be the oriented edge matrix.

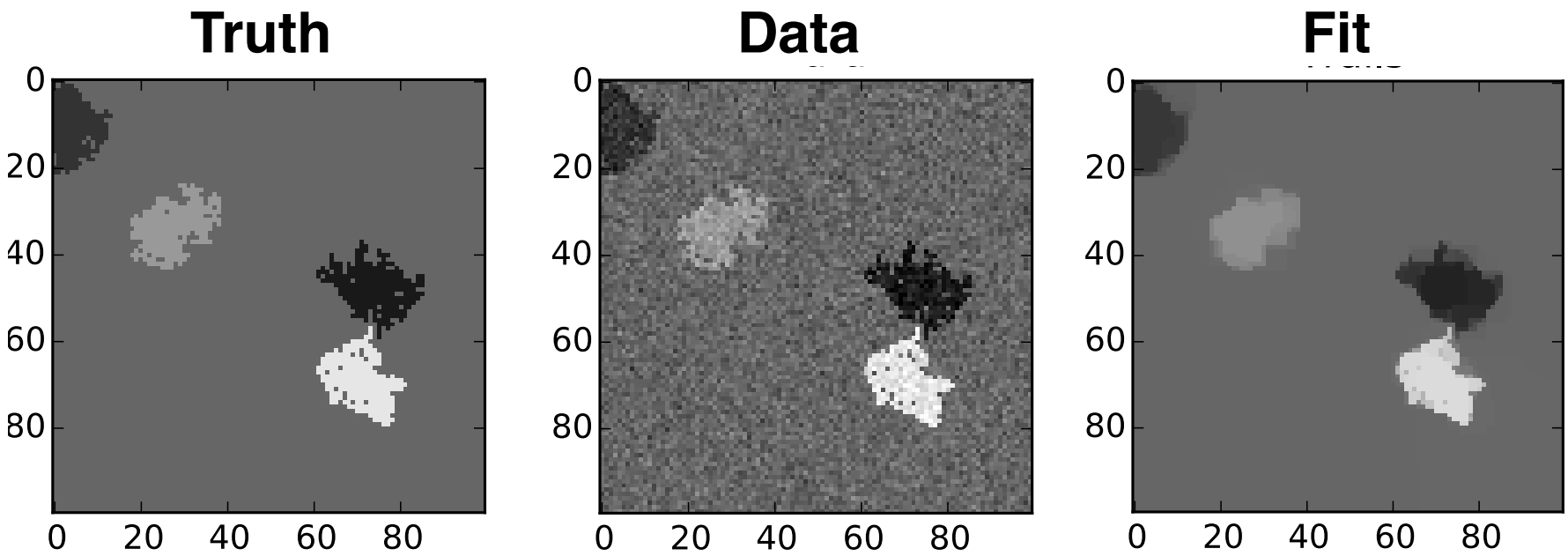


$$\begin{pmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1
\end{pmatrix}$$

Graph fused lasso: given data y_s at each vertex,

$$\underset{\beta \in \mathcal{R}^N}{\text{minimize}} \quad \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|D\beta\|_1 ,$$

Graph fused lasso



Graph trend filtering

As in the 1D case, we can recursively define higher-order versions of D to yield the graph trend-filtering estimator.

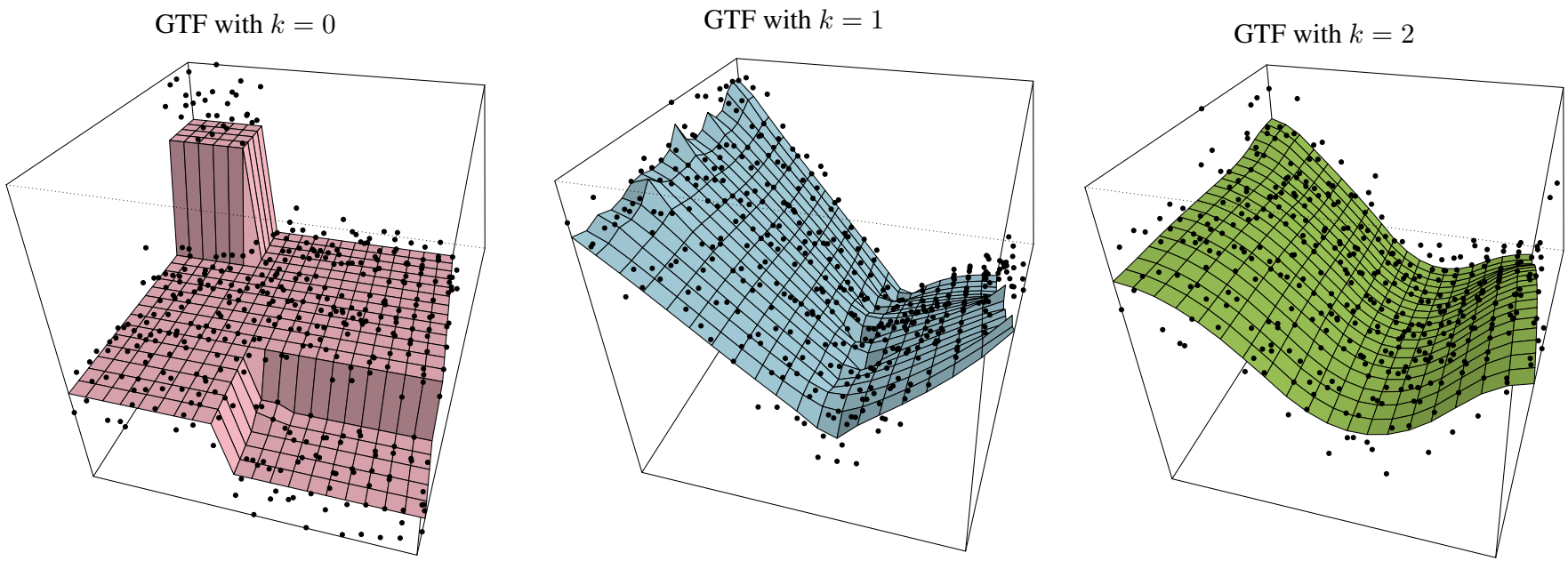


Figure from Wang et al (2015).

Some data-analysis examples

1) Estimating a spatially varying density function
   - work led by Wesley Tansey

2) Spatially aware multiple testing
   - work led by Wesley Tansey

3) Density estimation and deconvolution
   - work led by Oscar Padilla

Problem 1: spatially aware anomaly detection

The goal: detect gamma radiation anomalies

- Monitor a port or a border crossing
- Find a lost source (e.g. a medical radio-isotope at a hospital).
- Detect a radiological dispersal device (a dirty bomb).

The setup:

- Small cesium-iodide detector (on officer, in vehicle, etc.)
- Detector yields photon counts x_t across 4096 channels
- Detector hooked up to Raspberry Pi + iPhone that continuously compares x_t versus the background f_0.

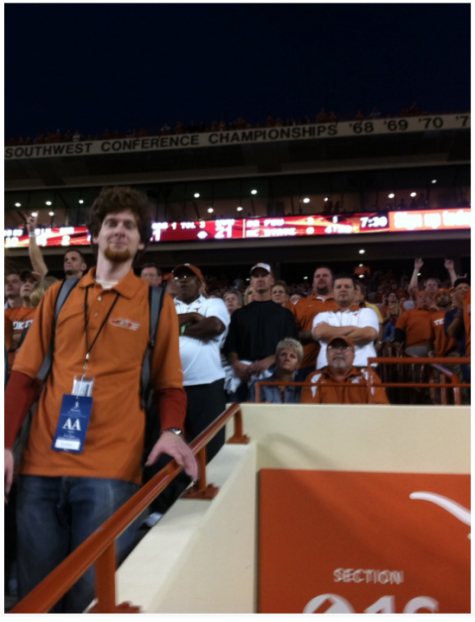
The statistical problem: what model is x_t from?

$$H_0^{(t)} : x_t \sim f_0 \quad \text{versus} \quad H_{\text{☢}}^{(t)} : x_t \sim \text{other}$$

Our data


mophie
BRIDGEPORT


YoYo! LTE 12:13 PM
192.168.20.6:8080
41 cps
Histogram


SOUTHWEST CONFERENCE CHAMPIONSHIPS '68 '69 '70
SECTION


LONGHORNS

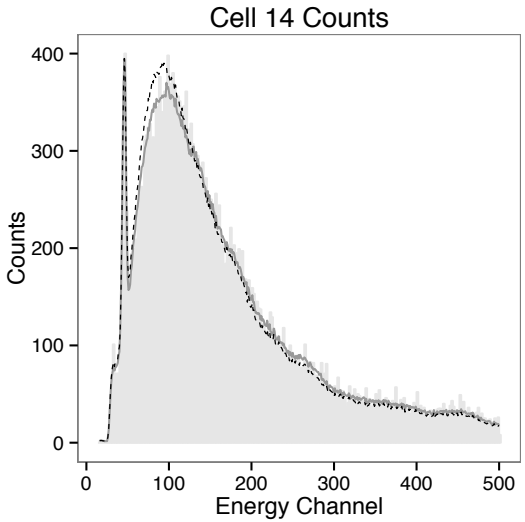
Our data

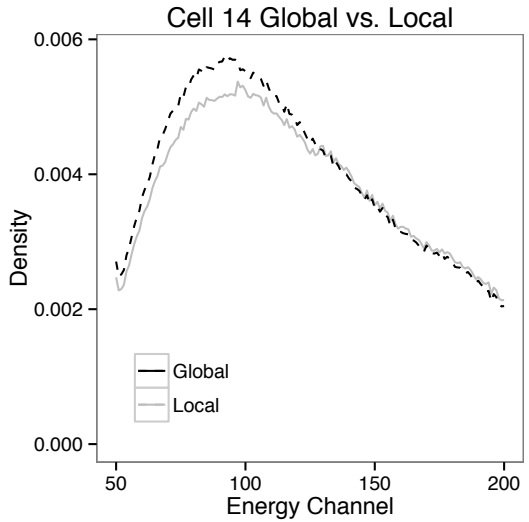

POLICE
2850
THE UNIVERSITY OF TEXAS AT AUSTIN

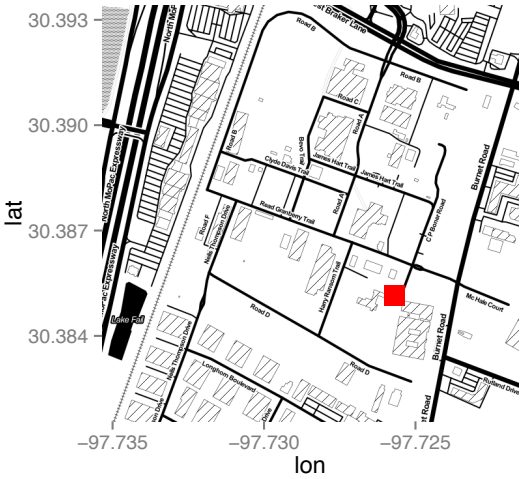

POLICE
714
THE UNIVERSITY OF TEXAS AT AUSTIN

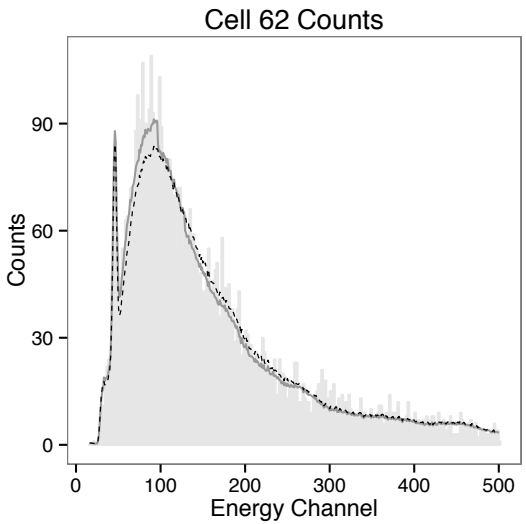

SET YOUR CARTS 5 FEET APART
5 feet
RECYCLING

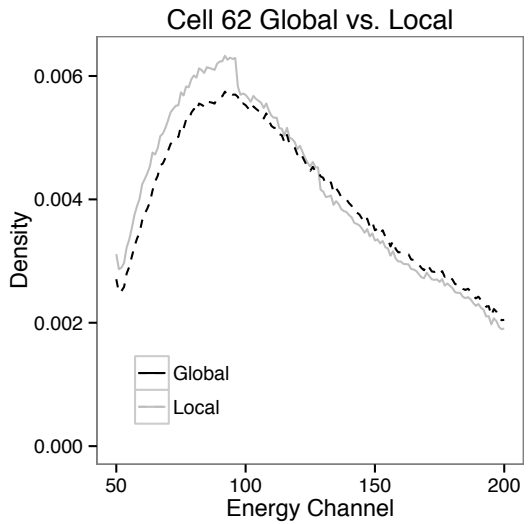
Spatial variation in spectrum











Two different locations.

The whole pipeline

Instrument calibration:

- Cheap ($\approx$ \$5000) detectors allow us wider coverage but are noisier (temperature, rain, instrument-level variability).
- Moderate amount of data available.

Background mapping:

- "Sharp + smooth," both in spectral and spatial dimensions
- Lots of data, unevenly distributed over monitoring area

Anomaly detection: lots of structure and prior knowledge

- $^{40}\mathrm{K}$: fertilizer
- $^{99}\mathrm{Tc}$: medical imaging
- $^{137}\mathrm{Cs}$, $^{60}\mathrm{Co}$, $^{192}\mathrm{Ir}$: call the FBI
- $^{234}\mathrm{U}$, $^{239}\mathrm{Pu}$: call the Air Force

Multiscale spatial density smoothing

The idea: motivated by Pólya trees (c.f. Hanson and Yang, 2007).

- **Split** into sub-problems via recursive partitioning.
- **Smooth** the half-space probabilities over the spatial lattice using binomial graph trend filtering.
- **Merge** the smoothed probabilities to yield $\hat{f}_0^{(s)}$, $s \in \mathcal{V}$.
- Reserve the power/hassle/expense of full Bayes analysis for the other parts of the pipeline.

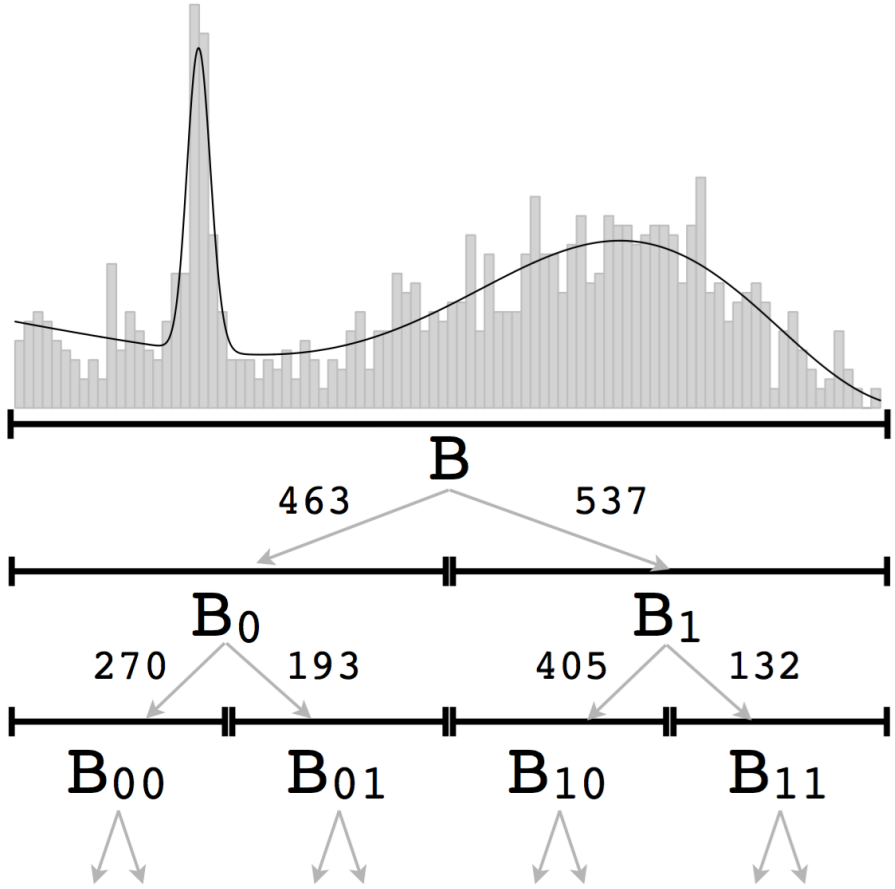
Notable points:

- Reduces the functional smoothing problem to a set of embarrassingly parallel scalar smoothing problems.
- Extremely fast and scalable to very large data sets (dominant cost = loading data into memory).

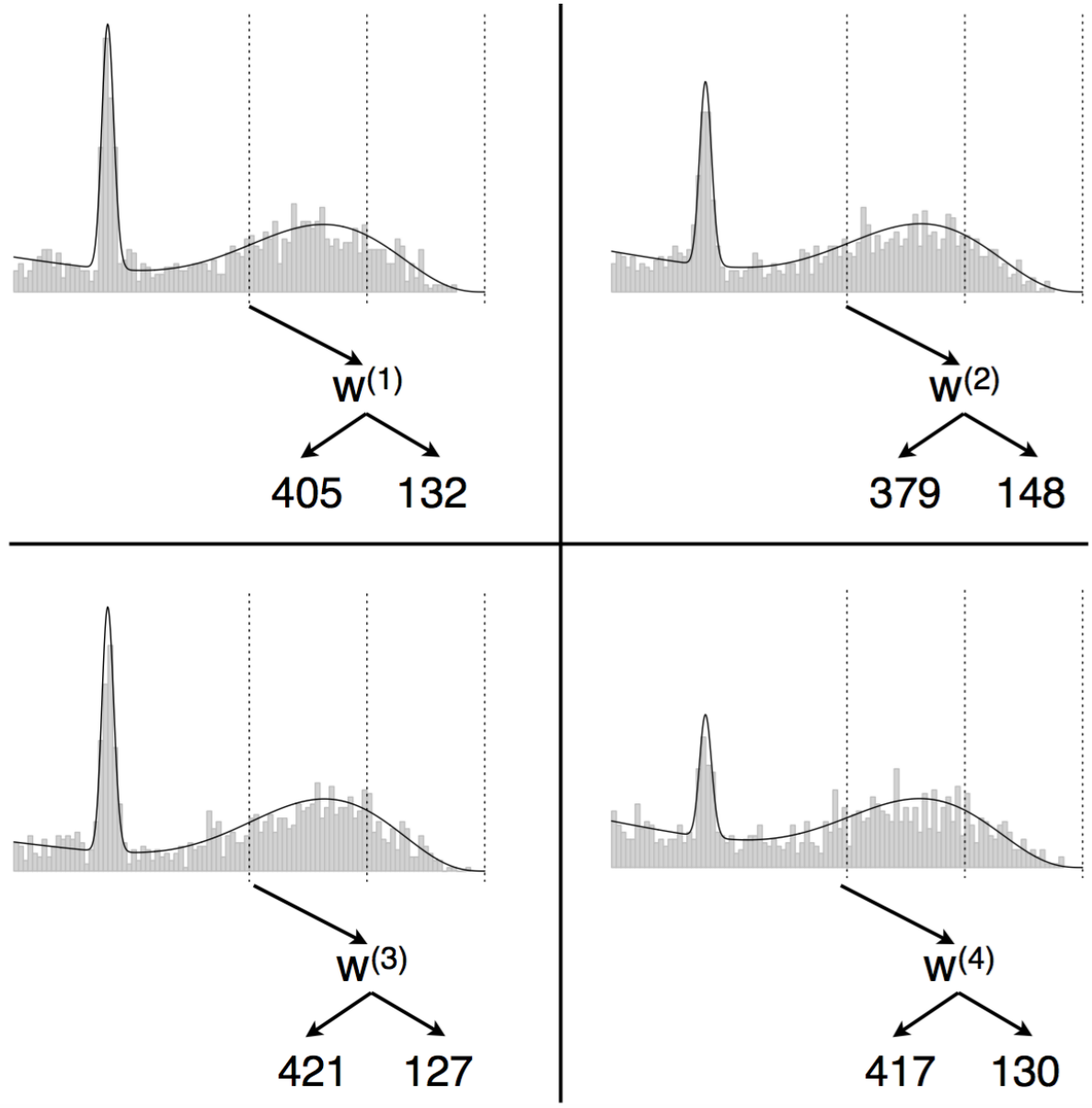
Recursive dyadic partitions

Let $(x_1, \ldots, x_n)$ be a sample from $f(x)$.

- n_γ: number of samples in the parent set B_γ.
- $y_{\gamma 0}$: number of samples in the left child set $B_{\gamma 0}$.



Spatial variation: a 2x2 example



Spatial variation

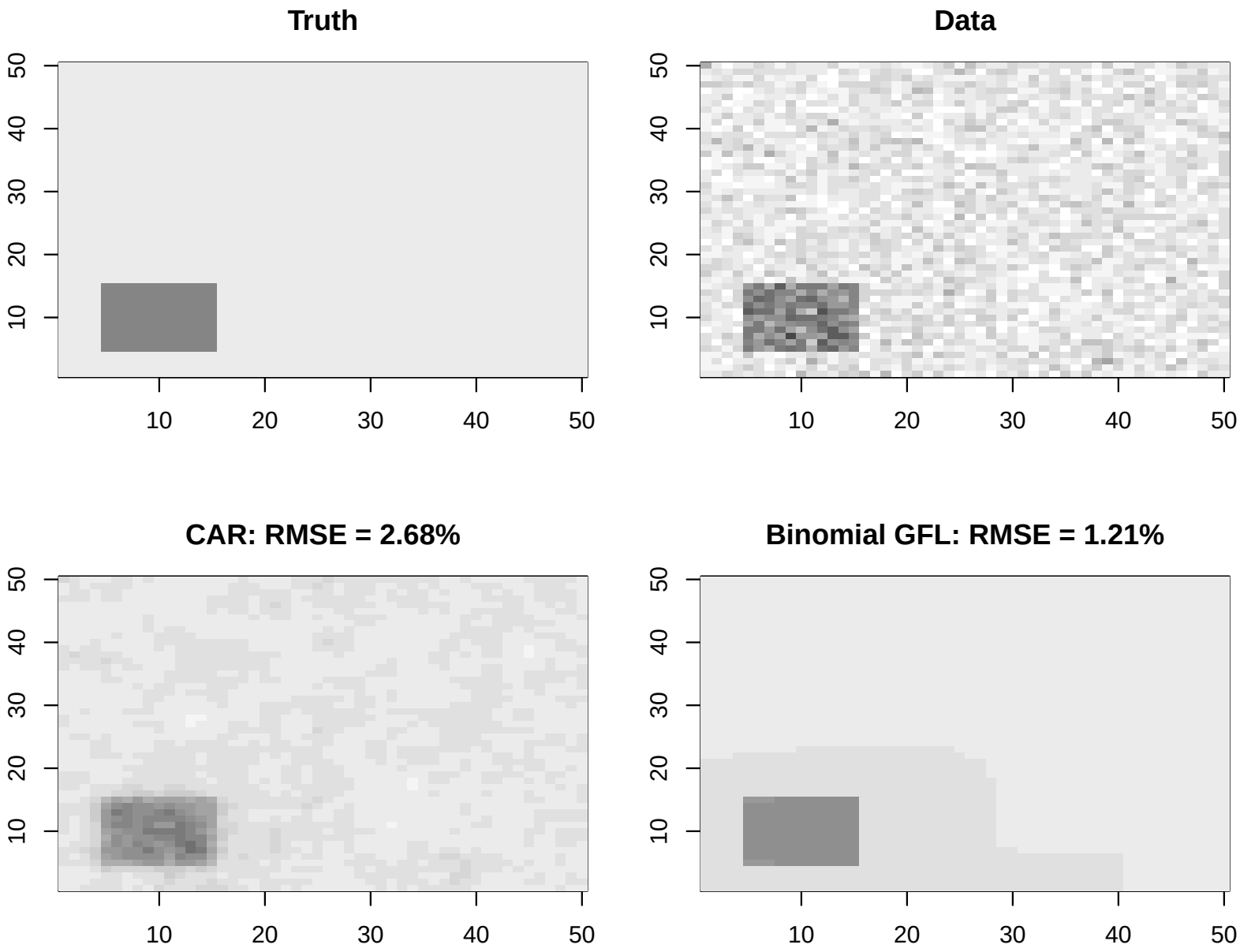
Now consider a specific split in the tree and drop the γ index. We want to estimate the "left-child" splitting probability across all spatial sites in our graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

$$y^{(s)} \sim \text{Binom}\left(n^{(s)}, \frac{e^{\beta^{(s)}}}{1 + e^{\beta^{(s)}}}\right), \quad s \in \mathcal{V}$$

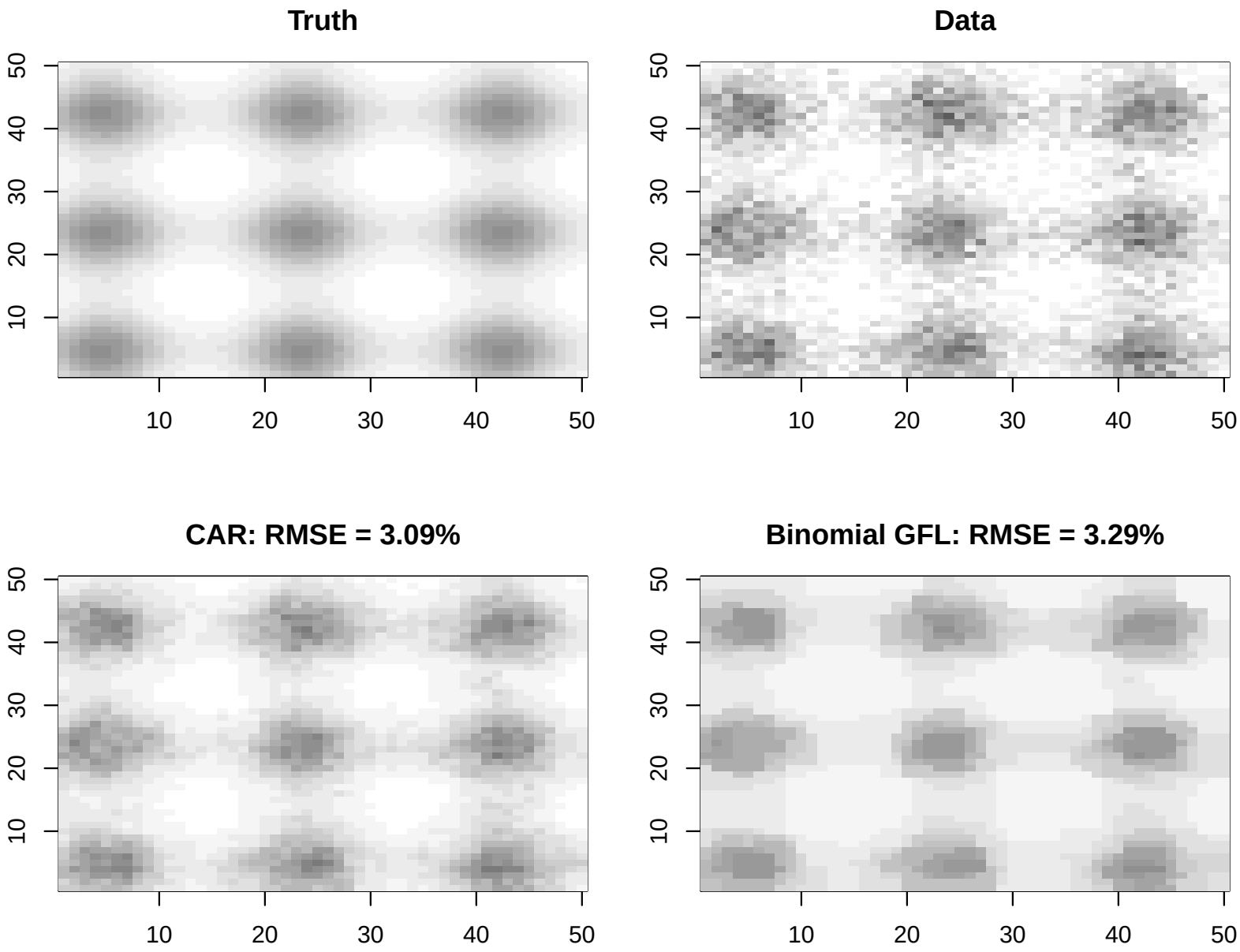
We enforce spatial smoothness by solving the following optimization problem for all splitting nodes, in parallel:

$$\underset{\beta \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{s \in \mathcal{V}} \left\{ n^{(s)} \log\left(1 + e^{\beta^{(s)}}\right) - y^{(s)} \beta^{(s)} \right\} + \lambda \|D\beta\|_1$$

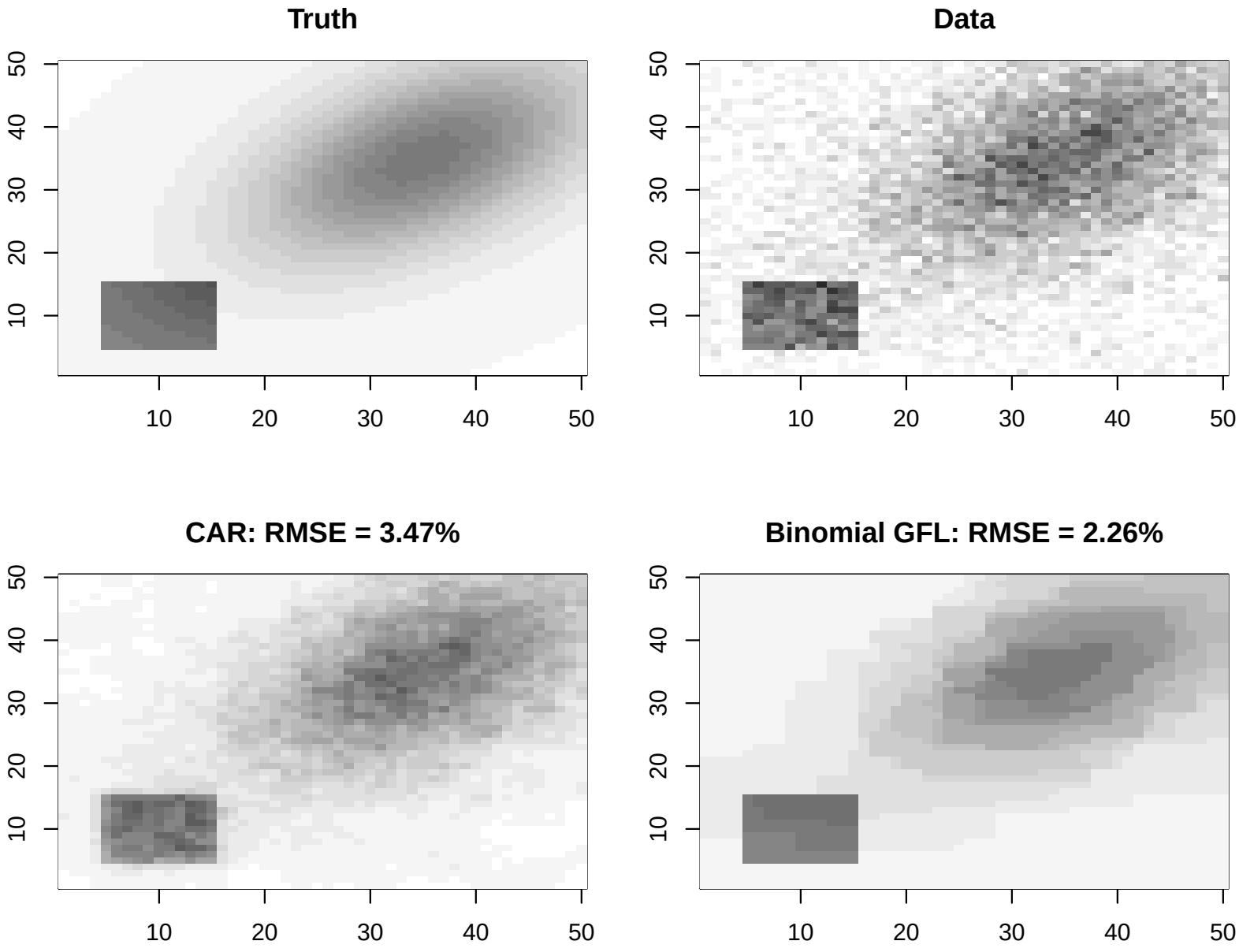
Smoothed splitting probabilities: plateaus



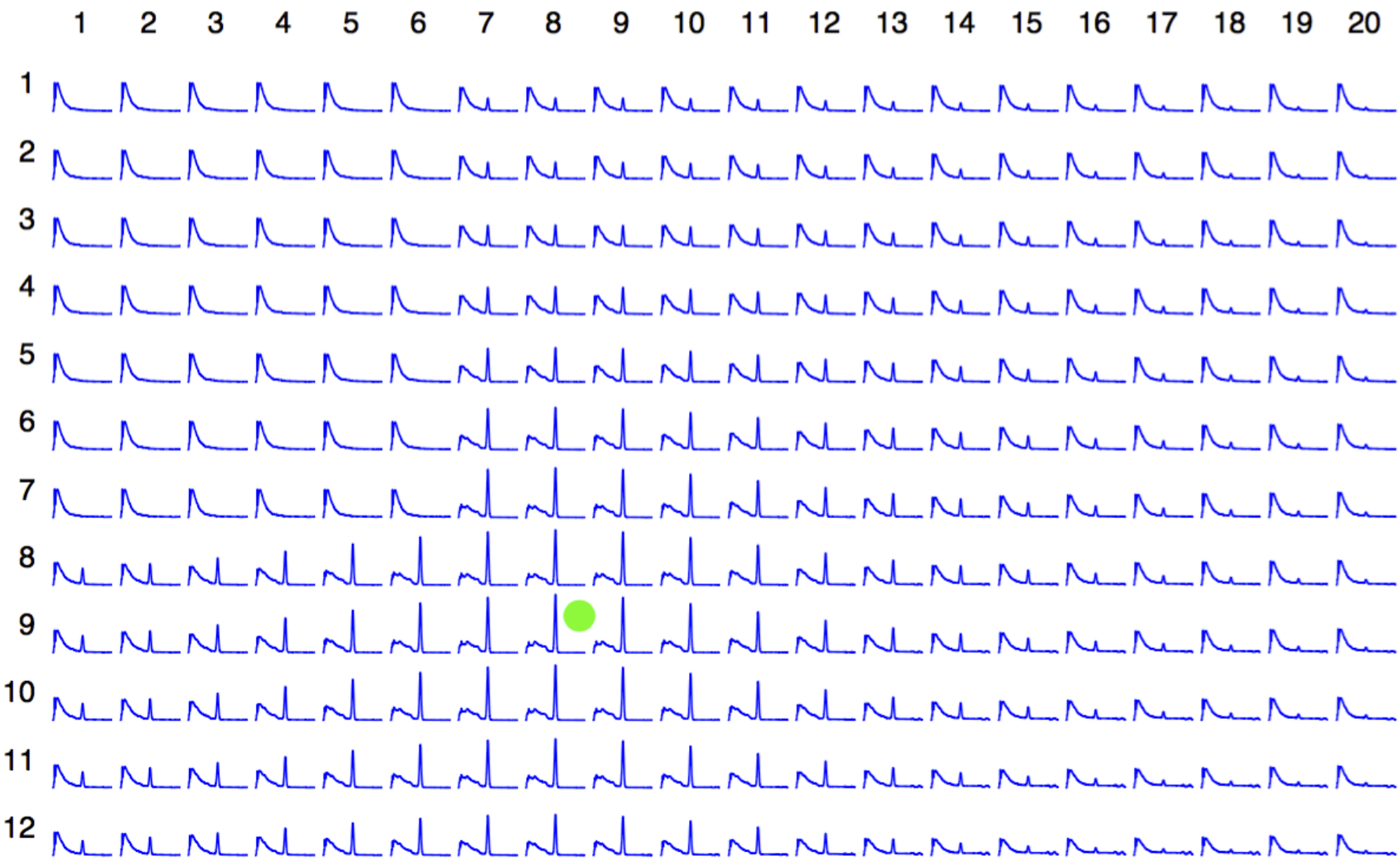
Example 2: sine wave



Example 3: smooth + jump



A benchmarking experiment



A benchmarking experiment

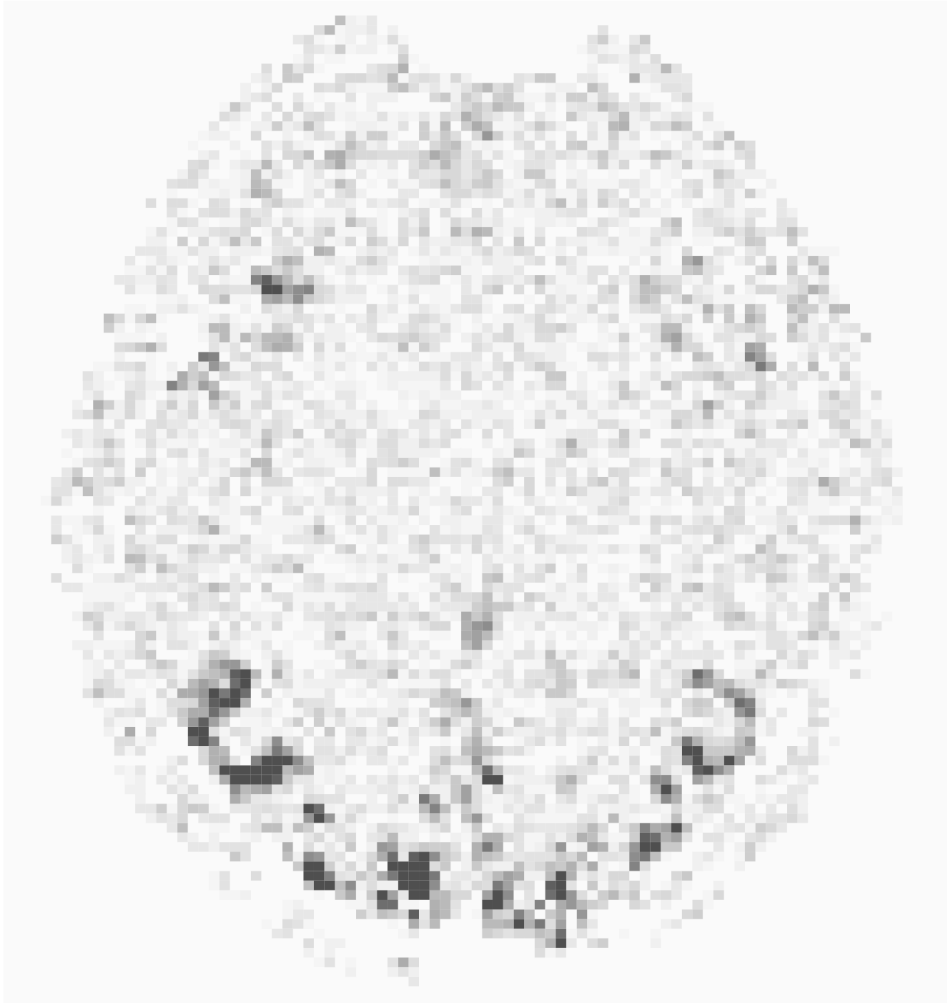
Dwell (s)	Histogram	Haar-Fisz	Laplacian	Multiscale
5	1.64	1.64	0.39	0.18
10	1.10	1.10	0.33	0.13
15	0.83	0.83	0.51	0.12
20	0.70	0.70	0.36	0.10
30	0.56	0.56	0.25	0.09
60	0.40	0.40	0.24	0.07
120	0.27	0.21	0.24	0.06
300	0.17	0.11	0.23	0.05

Total variation distance $\times 10^2$

Table: Total-variation distance between the estimated and true densities, averaged across all grid cells. Dwell: background observation time (s).

Problem 2: spatially dependent multiple testing

Raw z scores from a single horizontal section



fMRI experiment on working memory.

Data from Poldrack et. al (2012).

Problem 2: spatially dependent multiple testing

This is archetypal of many modern multiple-testing problems.

1. The data: many subjects, each with test statistic z_i
2. The goal: find which z_i correspond to signals
3. The challenge: control for multiple testing
4. The extra wrinkle: spatial structure

Existing procedures are pretty good at handling (1)–(3).

But not for dealing with (4) in a way that is:

- robust
- scalable
- principled.

The two-groups model

Suppose that the test statistics $z_1, \ldots, z_N$ arise from the mixture

$$z \sim c \cdot f_1(z) + (1 - c) \cdot f_0(z),$$

where $c \in (0, 1)$, and where f_0 and f_1 the null ($h_i = 0$) and alternative ($h_i = 1$) distributions of the test statistics.

The "Bayes oracle" posterior probability that $h_i = 1$ is

$$w_i = \frac{P(h_i = 1) f_1(z_i)}{f(z_i)} = \frac{c \cdot f_1(z_i)}{c \cdot f_1(z_i) + (1 - c) \cdot f_0(z_i)}.$$

The empirical-Bayes approach: use plug-in estimates.

$$\hat{w}_i = \hat{\mathrm{P}}(h_i = 1 \mid z_i) = \frac{\hat{c} \cdot \hat{f}_1(z_i)}{\hat{c} \cdot \hat{f}_1(z_i) + (1 - \hat{c}) \cdot f_0(z_i)}.$$

FDR smoothing

Suppose now that we have spatial information: each z_i is associated with a site s_i on a graph.

In fMRI, each s_i is a voxel, and $\mathcal{G}$ encodes voxel adjacencies.

We assume that the prior probabilities are site-dependent:

$$z_i \sim c_i \cdot f_1(z_i) + (1 - c_i) \cdot f_0(z_i) \quad (2)$$

$$c_i = \frac{e^{\beta_i}}{1 + e^{\beta_i}}. \quad (3)$$

FDR smoothing

FDR smoothing estimates the spatial field of log odds $\beta_1, \ldots, \beta_N$.

It is not a full Bayes analysis (an fMRI scan: ≈ 1 million nodes). It makes compromises for the sake of feasible computing times.

More like empirical Bayes:

- We do get posterior probabilities for individual hypotheses:

$$\hat{\mathrm{P}}(h_i = 1 \mid z_i) = \frac{\hat{c}_i \cdot \hat{f}_1(z_i)}{\hat{c}_i \cdot \hat{f}_1(z_i) + (1 - \hat{c}_i) \cdot f_0(z_i)}, \quad \hat{c}_i = \frac{1}{1 + e^{\beta_i}}$$

- We do not get a posterior distribution for $\beta_1, \ldots, \beta_N$.

FDR smoothing

Let $l(\beta)$ be the negative log-likelihood for the two-groups model:

$$l(\beta) = -\sum_{i=1}^{n} \log\left[\left(\frac{e^{\beta_i}}{1+e^{\beta_i}}\right) f_1(z_i) + \left(\frac{1}{1+e^{\beta_i}}\right) f_0(z_i)\right].$$

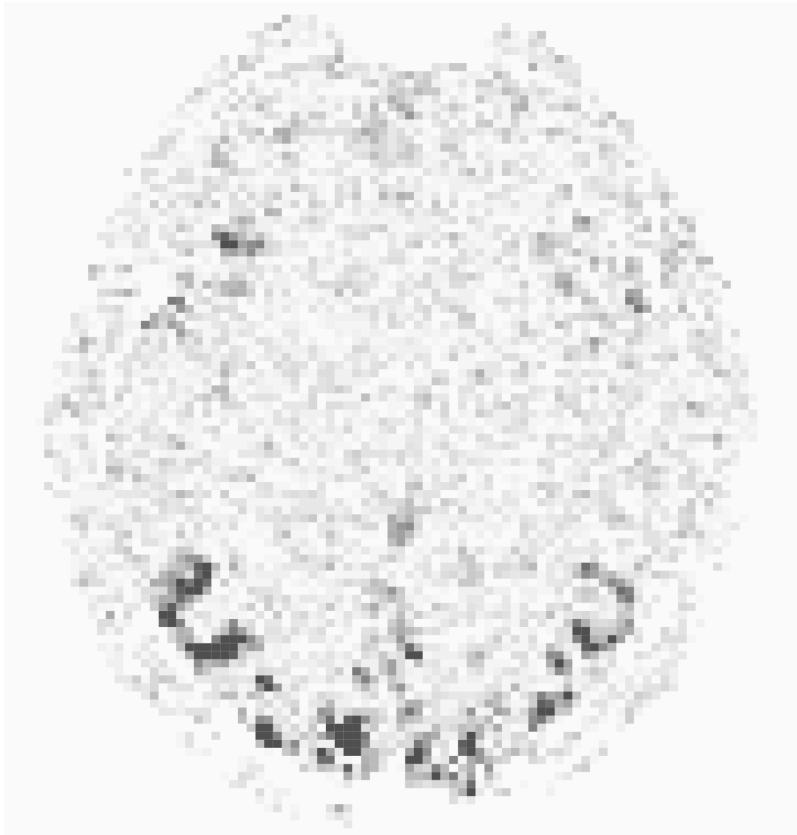
We learn the prior by solving the following optimization problem:

$$\underset{\beta \in \mathbb{R}^n}{\text{minimize}} \quad l(\beta) + \lambda \|D\beta\|_1$$

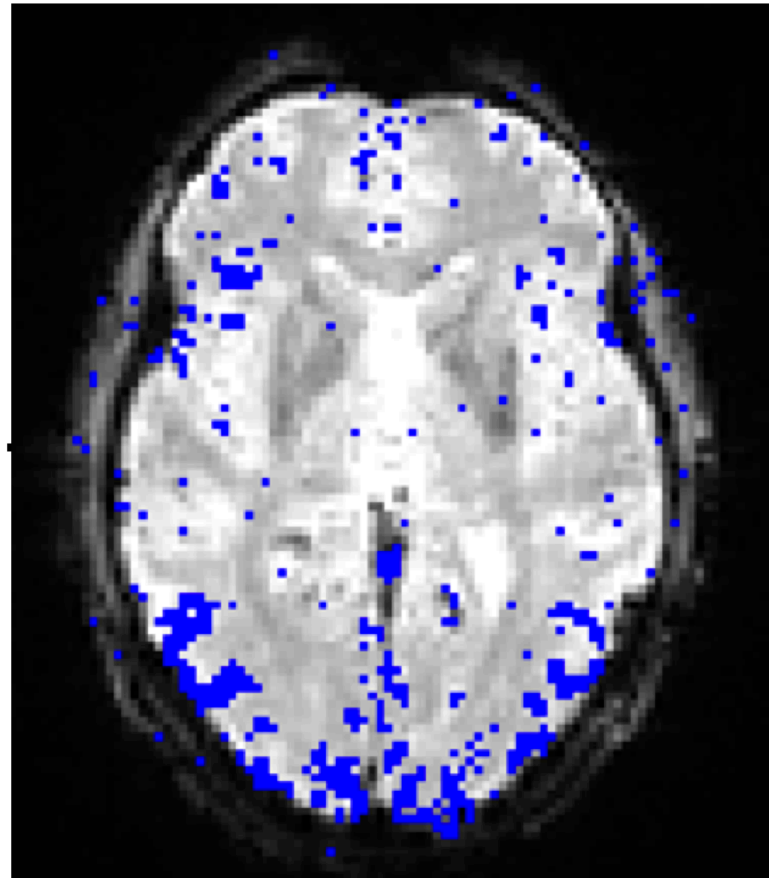
Here D is the familiar discrete-difference matrix.

FDR smoothing: fMRI example

Raw z scores from a single horizontal section



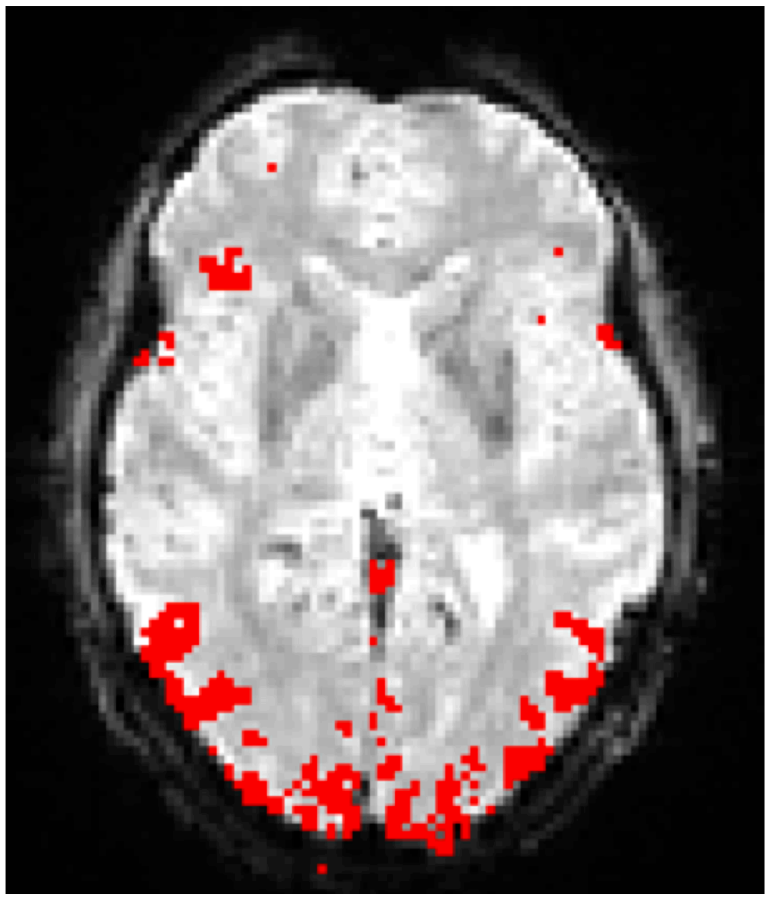
Findings using the Benjamini-Hochberg method



FDR smoothing: fMRI example

Estimated local fraction of signals

Findings using FDR smoothing



Summary

FDR smoothing:

- automatically finds localized regions of significant tests.
- improves power.
- yields a Bayesian posterior probability for each test.
- still controls FDR at the nominal level.
- is very fast.

Other applications:

- tests along the chromosome
- screening in environmental sensor networks
- networks derived from gene ontologies
- spatiotemporal FMRI
- . . . .

Problem 3: deconvolution

Suppose that we observe $\mathbf{y} = (y_1, \ldots, y_n)$ from

$$y_i \mid \mu_i \sim \phi(y_i \mid \mu_i), \quad \mu_i \overset{i.i.d.}{\sim} f_0,$$

where f_0 is an unknown mixing distribution (ϕ known).

Two possible inferential goals:

- Estimate the mixing distribution f_0 (deconvolution).
- Estimate the normal means μ_i.

Tweedie's formula (Robbins, 1956; Efron, 2010). For ϕ Gaussian,

$$E(\mu \mid y) = y + \frac{\mathrm{d}}{\mathrm{d}y} \log m(y) = y + \frac{m'(y)}{m(y)},$$

where $m(y) = \int p(y \mid \mu)\ f_0(\mu) d\mu$ is the marginal density.

Deconvolution

The classical estimator is from Kiefer and Wolfowitz (1956):

$$\hat{f} = \arg\max_{f \in \mathcal{F}} \prod_{i=1}^{n} m_f(y_i),$$

where $m_f(y) = \int \phi(y - \mu) df(\mu)$ is the marginal.

The KW estimator has some appealing features:

- completely nonparametric
- no tuning parameters, translation invariant
- consistent under fairly general conditions

But: $\hat{f}$ is a discrete distribution involving as many as $n + 1$ point masses (a "Dirac catastrophe").

Deconvolution: other approaches

Geman and Hwang (1982): method of sieves. **Provably consistent only for the marginal.**

Various other methods based on kernels and penalized likelihood.

Dirichlet process:

$$(\mu_i \mid \theta_i, \tau_i^2) \sim \mathrm{N}\left(\theta_i, \tau_i^2\right)$$
$$(\theta_i, \tau_i^2) \sim f\,, \quad f \sim \mathrm{DP}\left(\alpha, G_0\right),$$

Concentration rates: Donnet et al. (2014). Related models: Do et al. (2005) and Muralidharan (2010). **Does not scale well.**

Predictive recursion (Newton, 2002; Tokdar et al, 2009). **Very effective; order-dependent; opaque regularization.**

Our approach

A simple two-step "bin and smooth" procedure.

- "Bin" step: form a histogram of the sample with bin counts x_j
- "Smooth" step: form a surrogate Poisson likelihood for x_j and compute a MAP estimate of f_0 under a trend-filtering prior.

This simple nonparametric procedure yields excellent performance for deconvolution, at dramatically reduced computational cost versus full nonparametric Bayesian methods.

Our main theorems:

1. establish conditions under which the method yields a consistent estimate of the mixing distribution, and
2. provide finite-sample risk bounds for the estimator.

Our approach

The estimator is motivated by the variational problem

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & -\sum_{i=1}^{n} \log(\phi * f)(y_i) \\ \text{subject to} \quad & \int_{\mathcal{R}} f(\mu)\, d\mu = 1 \\ & \int_{\mathcal{R}} |\log f^{(k)}(\mu)|\ d\mu \leq t\,, \end{aligned}$$

If $k = 1$, this is an L_1 penalty on the score of f. Avoids Dirac catastrophes by shrinking toward a smooth estimate.

Our approach

Some helpful simplifications (theory still carries through):

- Penalize rather than constrain.
- Bin the data, yielding counts $x_1, \ldots, x_M$.
- Work with a Poisson rather than multinomial likelihood.

Thus our "bin-and-smooth" model assumes that

$$(x_j \mid \lambda_j) \sim \text{Poisson}(\lambda_j), \quad \lambda_j = \sum_{j=1}^{M} G_{ij} e^{\theta_i}.$$

where $G_{ij} = M\phi(\xi_i - \xi_j)$ is the Gaussian blur matrix (i.e. the discrete approximation to the convolution kernel).

The e^{θ_i}'s are proportional to the mixing density on a discrete grid.

Our approach

We estimate the mixing density by solving:

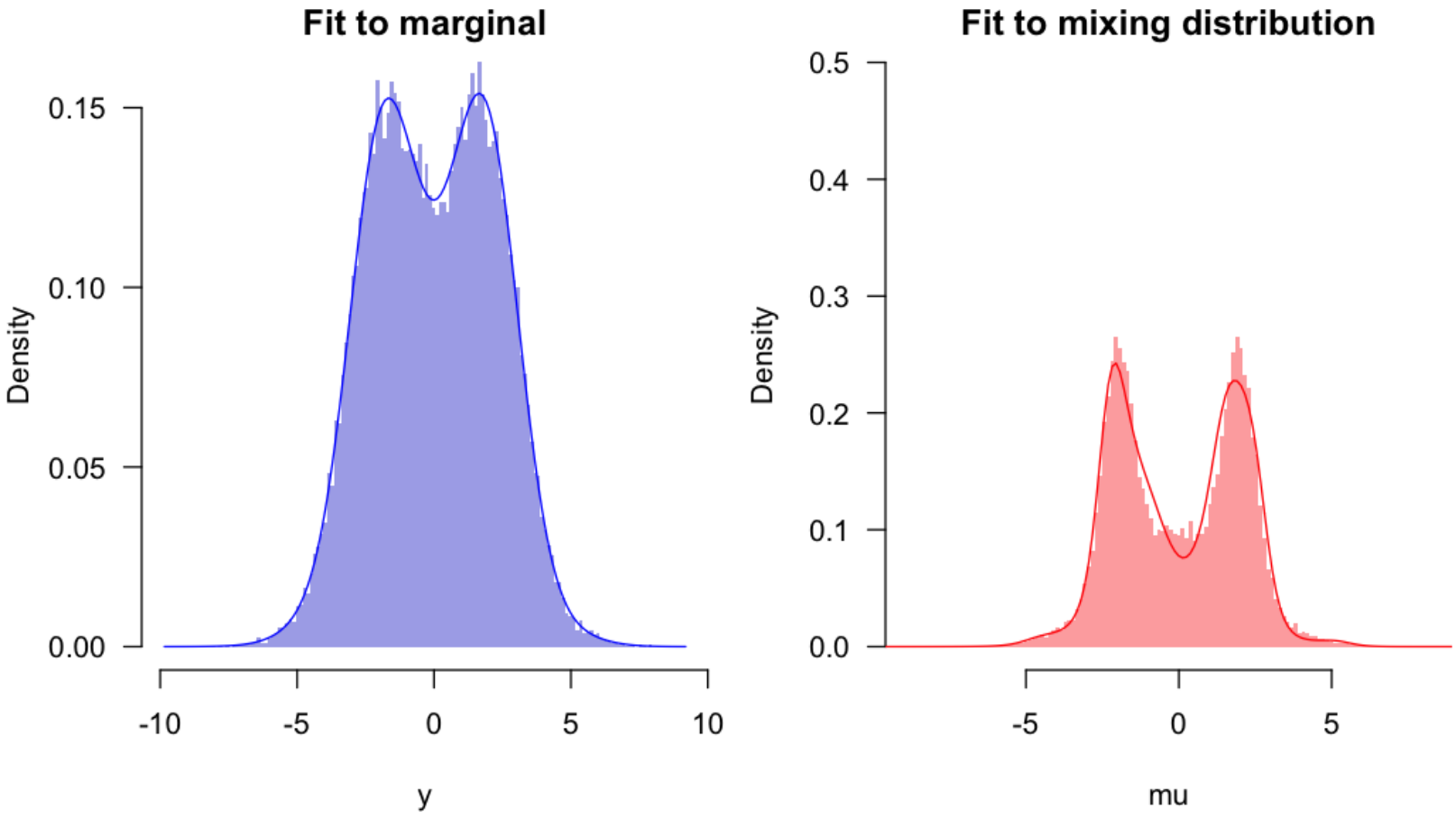
$$\underset{\theta \in \mathcal{R}^N}{\text{minimize}} \quad l(\theta) + \lambda \|D^{(k+1)}\theta\|_1 ,$$

where $D^{(k+1)}$ is the trend-filtering matrix. Solved by ADMM.

We have theorems for both the variational ($\hat{f}_V$) and discrete ($\hat{f}_D$) estimators. Under suitable conditions:

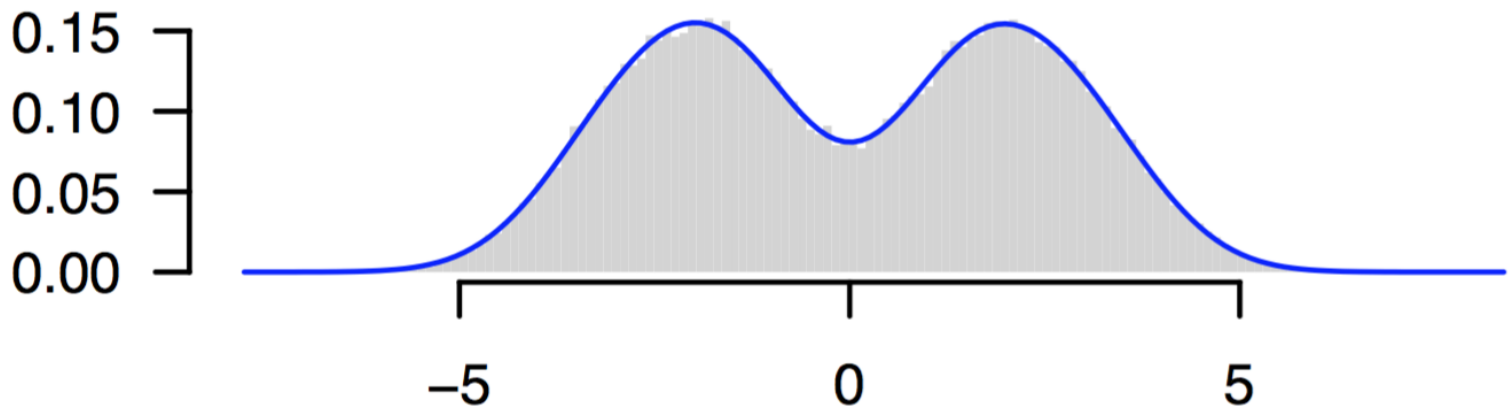
- The Hellinger distance between f and $\hat{f}_V$ satisfies a strong concentration inequality.
- The Kullback-Leibler divergence from f to $\hat{f}_D$ goes to 0 as n diverges (assuming $M \to \infty$ and $\lambda \to 0$ at appropriate rates).

The deconvolution path



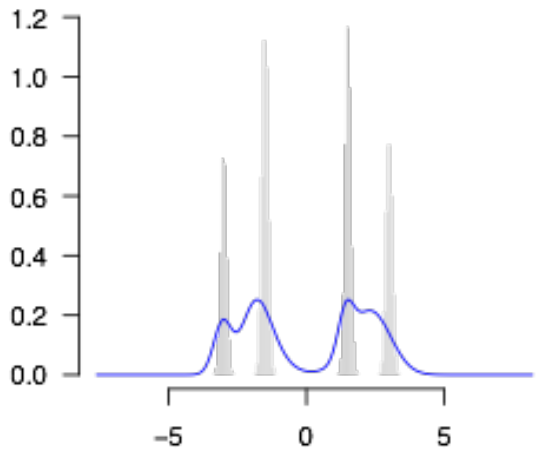
Examples

A harder example: the histogram and fitted marginal density.

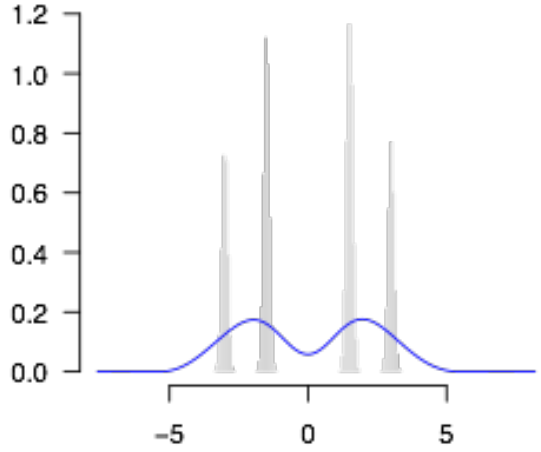


Examples

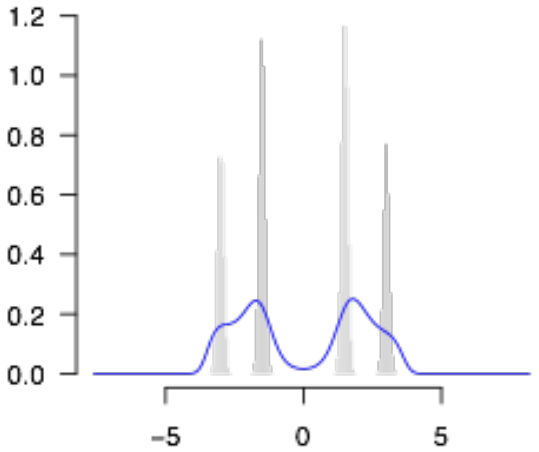
Here is the underlying f_0, together with the estimates from the best existing methods:



Dirichlet process mixture of normals



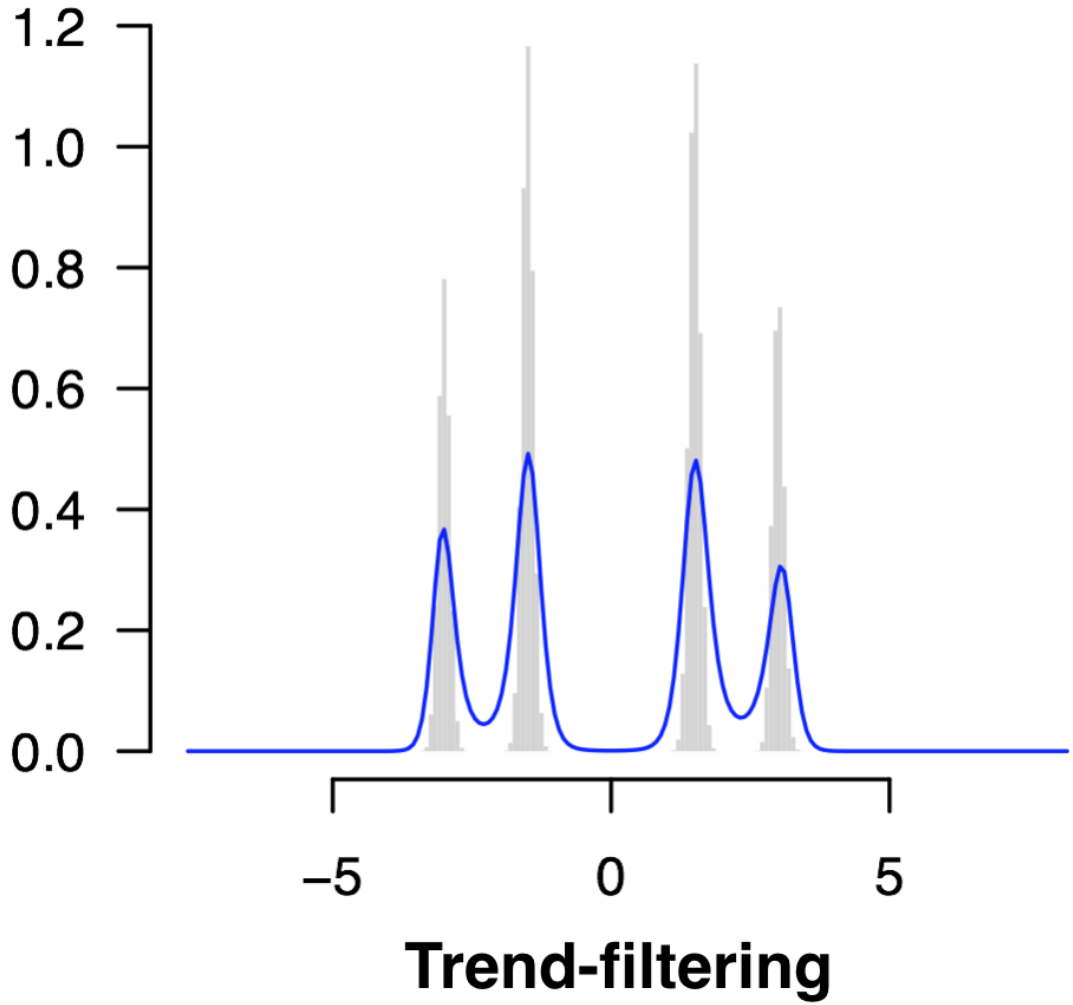
Fourier-transform kernel density estimate



Predictive recursion

Examples

Our method:



Summary

Discrete-difference smoothers:

- are wonderfully practical and versatile.
- scale to very large problems.
- adapt well to nonstationarity.

Summary

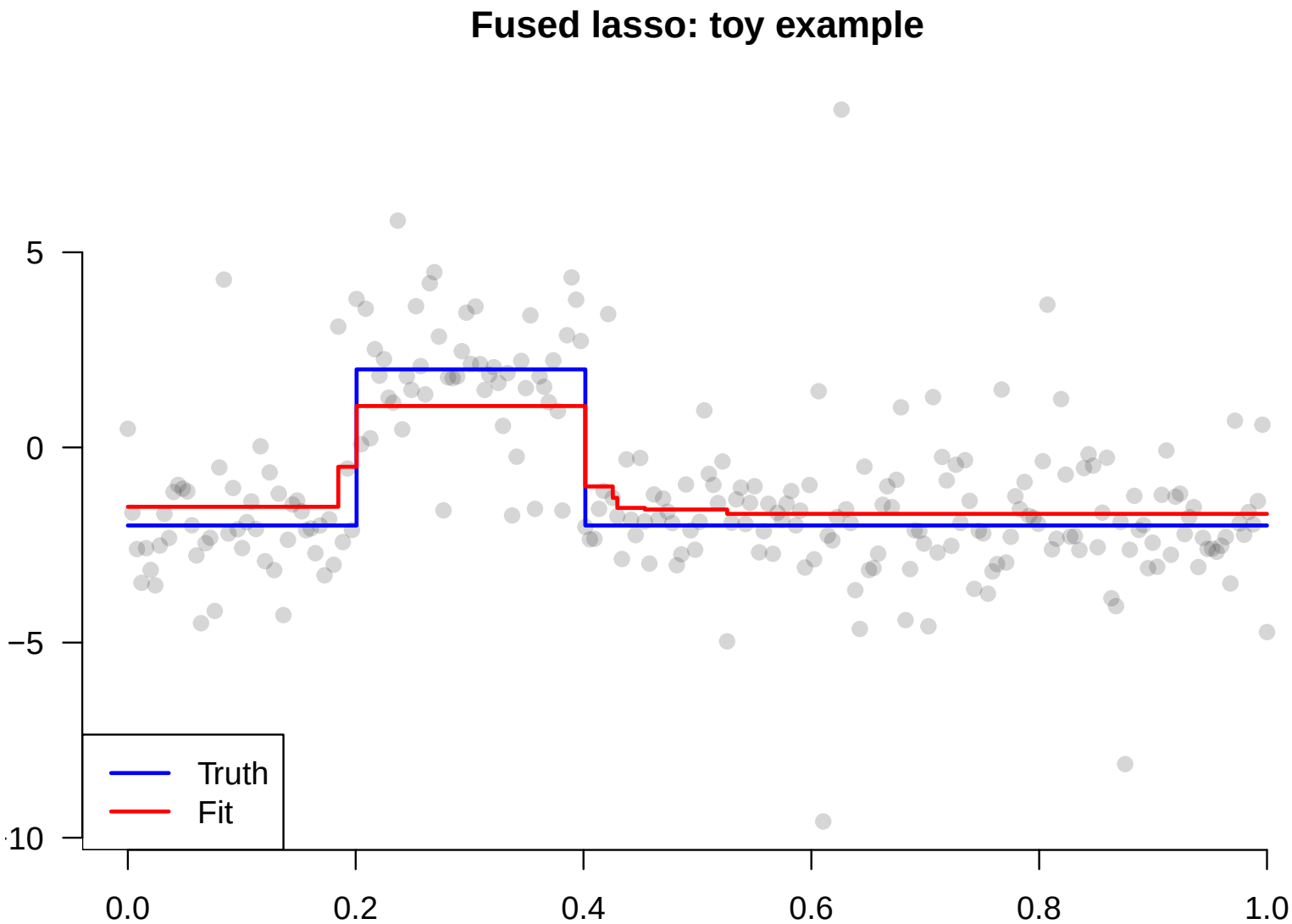
Discrete-difference smoothers:

- are wonderfully practical and versatile.
- scale to very large problems.
- adapt well to nonstationarity.

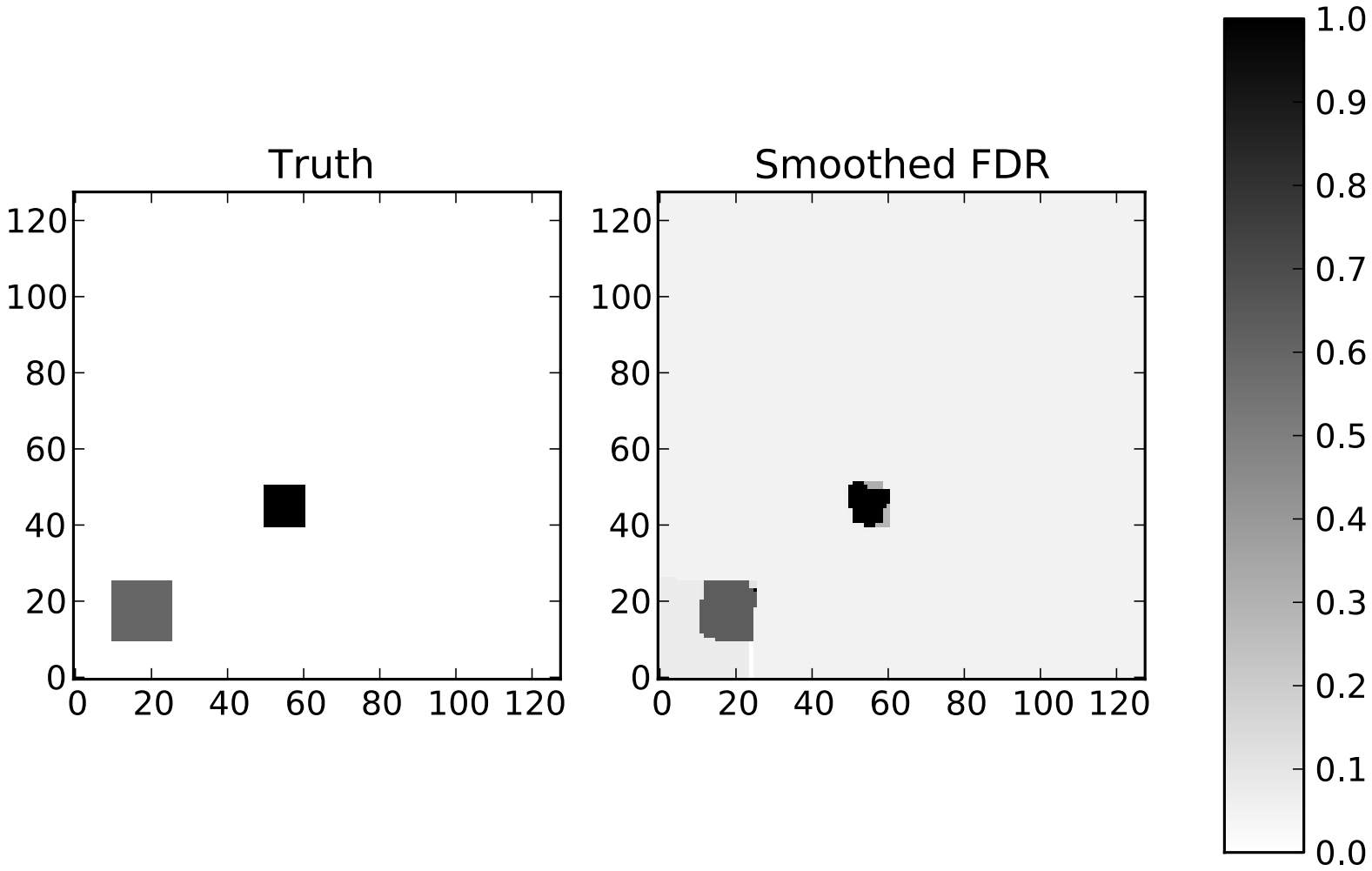
But there are still problems:

- The ℓ_1 penalty shrinks big jumps too much (sandpaper)
- Higher-order smoothers can be slow and unstable.
- The penalty parameter is difficult to choose outside squared-error loss.

Problem 1: non-diminishing bias (fused lasso)

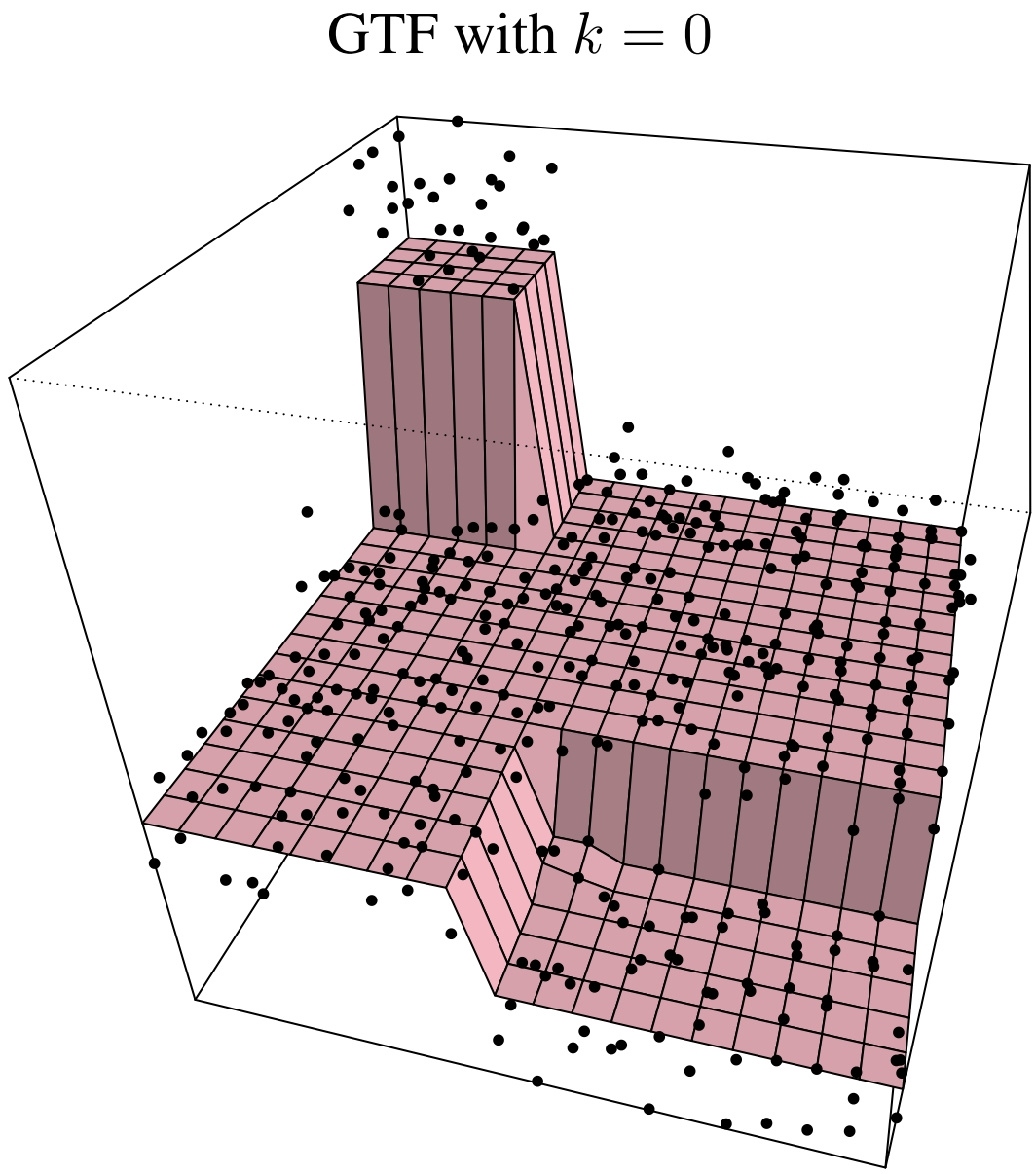


Problem 1: non-diminishing bias (FDR smoothing)



The true versus learned image of underlying prior probabilities under a nonparametric estimate of $f_1(z)$

Problem 1: non-diminishing bias (graph trend filtering)



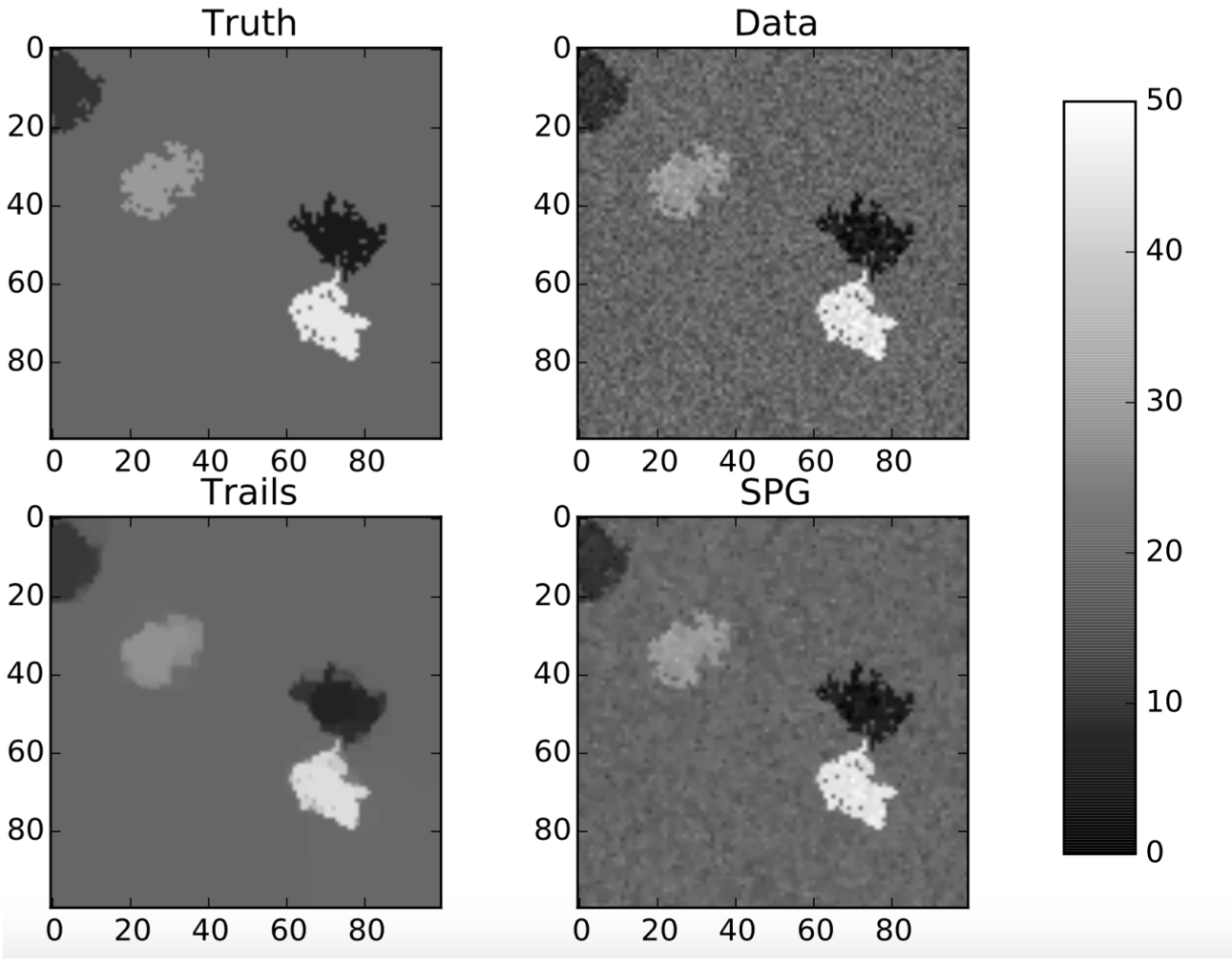
Problem 2: speed and stability beyond $k = 0$

A hierarchy of difficulty (best algorithm for each case):

- 1D fused lasso: **nearly instant** (taut string or DP)
- fused lasso on a grid graph: **very fast** (proximal stacking)
- 1D trend filtering, $k = 1$: **very fast** (ADMM)
- 1D trend filtering, $k = 2, 3$: **fast** (ADMM)
- fused lasso on other graphs: depends on degree distribution but **fast for most graphs** (trail decomposition or PMF)
- graph trend filtering, $k = 1$: can be unstable (proj. Newton)
- 1D trend filtering, $k \geq 4$: slow (ADMM)
- graph trend filtering, $k = 2$: very slow (ADMM)
- Graph trend filtering, $k \geq 3$: don't bother

A single family of models, but a wide variety of algorithms.

The importance of the right algorithm



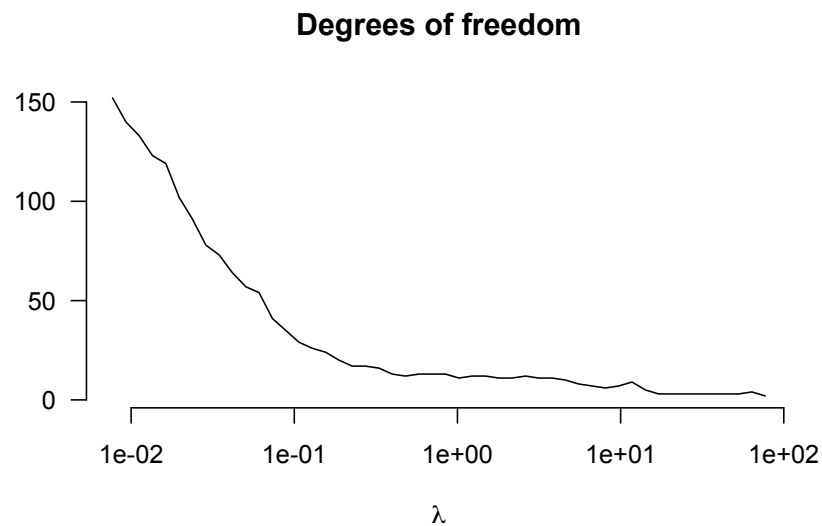
Eulerian decomposition proximal stacking (L) versus smoothed proximal gradient (R). Figure from Tansey and Scott (2015). Code at `www.github.com/tansey/gfl/`

Problem 3: choosing the hyperparameter

Issue 1: results for $\mathrm{df}(\hat{\beta}_\lambda)$ apply only to Gaussian case.

Issue 2: calculating even an ad-hoc approximation to $\mathrm{df}(\hat{\beta}_\lambda)$ is subject to numerical noise in many problems.

- Gaussian case: calculate $\mathrm{df}(\hat{\beta}_\lambda)$ by counting changepoints.
- But many algorithms smooth out these changepoints because they enforce exact zeros only at convergence



Bayes to the rescue

Let β be a signal on a graph. A Bayesian version of the GTF estimator would involve assuming the (improper) prior

$$p(\beta) \propto \exp\left\{-\lambda \|D\beta\|_1\right\},$$

If D is the zero-order smoother and we make the penalty ℓ_2:

$$\begin{aligned} p(\beta) &\propto \exp\left\{-\lambda \|D\beta\|_2^2\right\} \\ &= \exp\left\{-\lambda \beta D^T D \beta\right\} \\ &= \exp\left\{-\lambda \beta (R - A) \beta\right\} \end{aligned}$$

where (R, A) are the degree and adjacency matrices. Looks like an ICAR prior!

Bayes to the rescue

The Gaussian version:

$$p(\beta) \propto \exp\left\{-\lambda \|D\beta\|_2^2\right\}$$

An equivalent representation of the Bayes GTF prior is:

$$p(\beta \mid \Omega, \tau^2) \propto \exp\left\{-\frac{1}{2\tau^2}\beta^T \left(D^T \Omega D\right)\beta\right\}$$

$$\Omega = \text{diag}(\omega_j^2)$$

$$\omega_j^2 \sim \text{Exp}\,.$$

All conditionals have simple forms.

Existing work: Rouldes (2015) and Faulker and Minin (2015) consider the chain-graph case.

Horseshoe graph trend filtering

We consider a version that incorporates a horseshoe prior:

$$p(\beta \mid \Omega, \tau^2) \propto \exp\left\{-\frac{1}{2\tau^2}\beta^T\left(D^T\Omega D\right)\beta\right\}$$
$$\Omega = \text{diag}(\omega_j^2)$$
$$\omega_j \sim \text{C}^+(0,1)\,.$$

Very far from the first version of this idea:

- Geman and Reynolds (1992), Geman and Yang (1995) responding to the popularity of ROF (1992).
- Faulker and Minin, 2015.

Gibbs steps are fast: all matrices are structured and sparse (Laplacian linear systems can be solved in near-linear time).

Horseshoe graph trend filtering

Full Bayes inference: put a prior on τ and run MCMC. But. . .

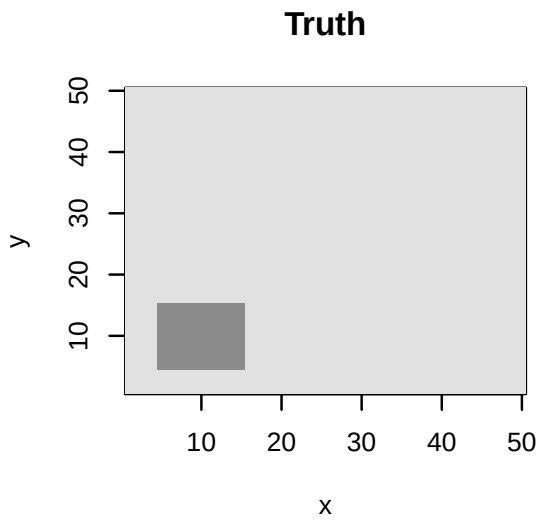
- Mixing over the global scale is very slow. (Gelman et. al., 2008 JCGS)
- And van der Pas et. al (2014) show that EB versions of the horseshoe prior can outperform the FB version from Carvalho, Polson, and Scott (2010).

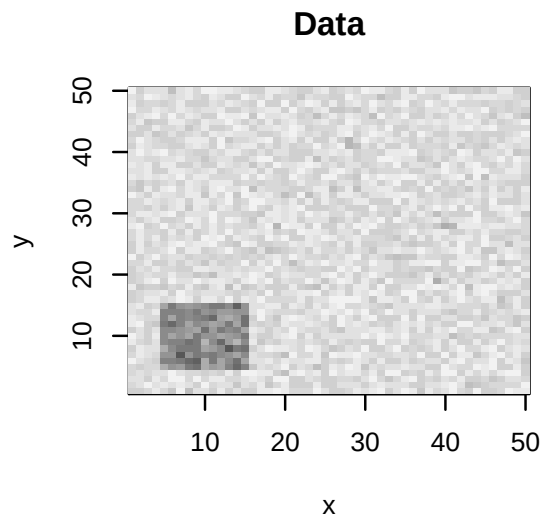
Our empirical-Bayes strategy:

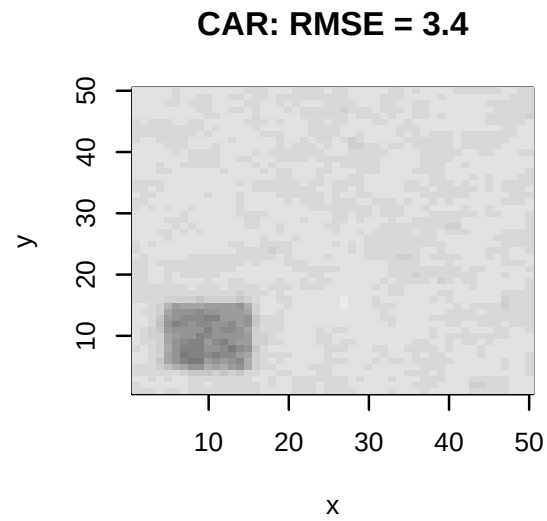
- Compute a Bayesian solution path: MCMC estimates in parallel for a grid of fixed τ values.
- Choose τ to minimize DIC.
- Side benefit: the worst mixing problems go away.

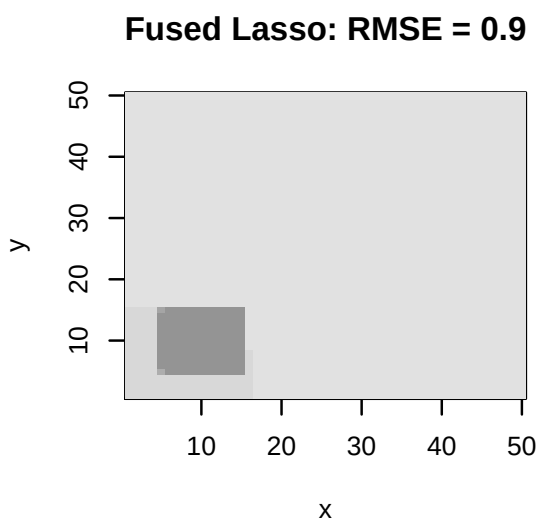
Following: several binomial examples on a 2D grid graph

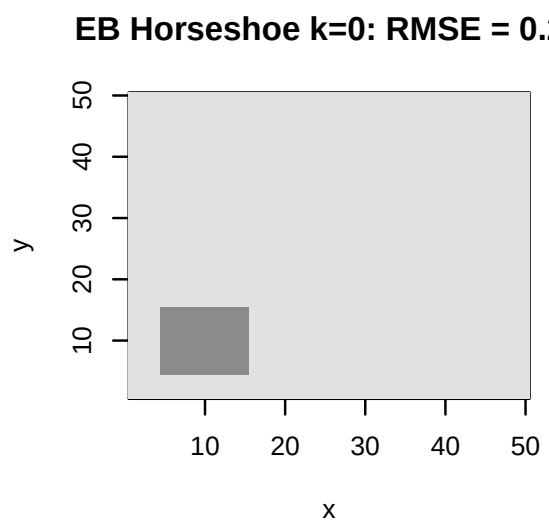
Binomial example 1: plateaus

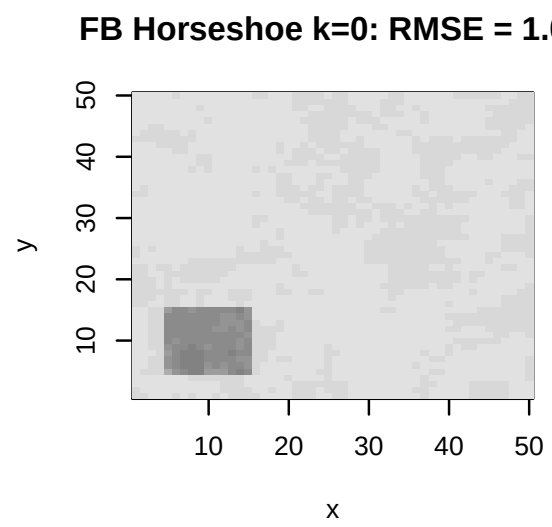




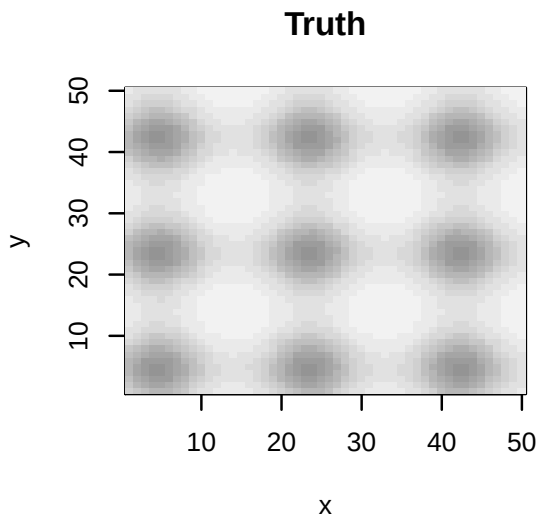


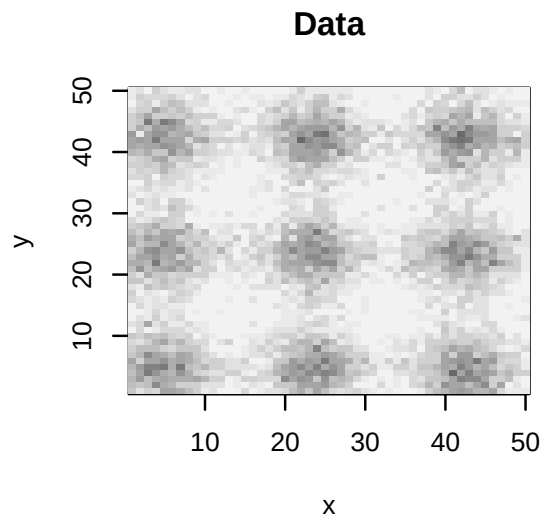


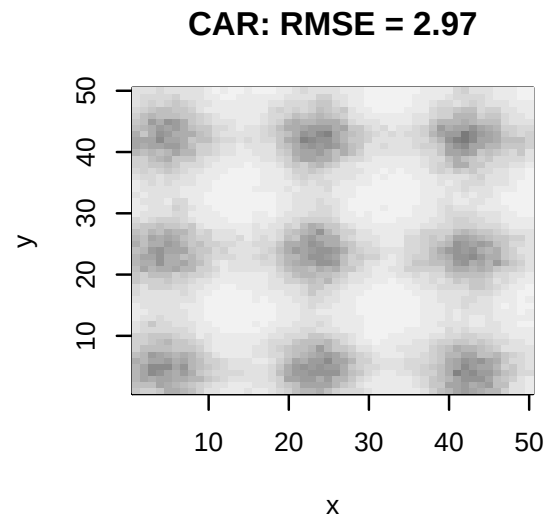


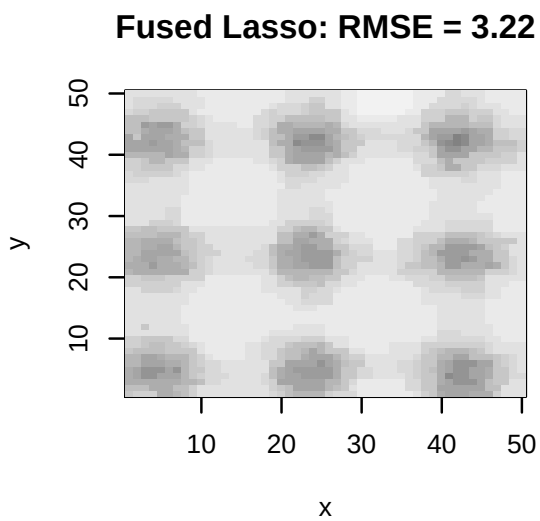


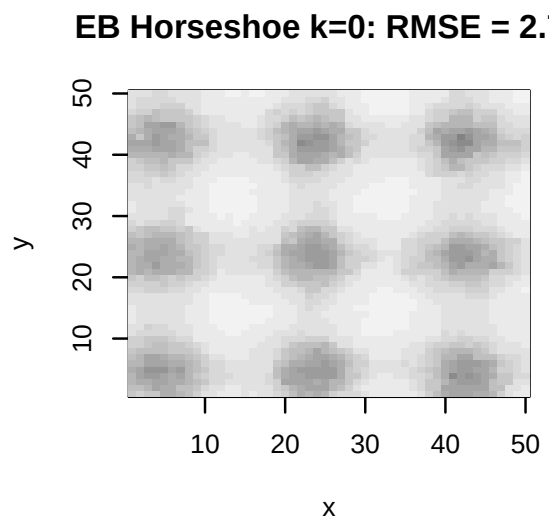
Binomial example 2: sine wave

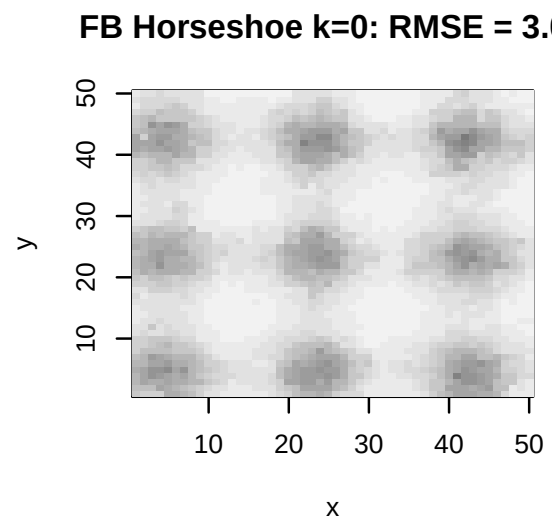












Conclusions

We should welcome these smoothers under big Bayesian tent. They:

- resemble familiar tools.
- can be used fruitfully as plug-in estimators in larger models/pipelines.
- handle "sharp + smooth" signals well.
- are very fast.

The Bayesian approach successfully addresses some practical obstacles posed by the classical estimator.

- Bayesian TF seem to outperform both the classical estimators and CAR models.
- Empirical Bayes seems to consistently outperform full Bayes, which noticeably undersmooths.

None of the Bayesian models are (yet) scalable enough to handle truly massive problems, where GFL still reigns.