Generalized Additive Functional Regression

David Ruppert

School of Operations Research & Information Engineering and Dept. of Statistical Science, Cornell University

Dec 5, 2011

Collaborators

- Mathew McLean, PhD student, Cornell University
- Giles Hooker, Assistant Professor, Cornell University
- Ana-Maria Staicu, Assistant Professor, North Carolina State University
- Fabian Scheipl, Postdoc, Ludwig Maximilian University of Munich

In functional regression either the response or at least one of the predictor variables is a function.

In this talk:

- Response is scalar
- There is one functional predictor, e.g., an EEG signal

Functional Linear Model (FLM)

A common approach is the functional linear model (FLM)

$$E(Y_i|X_i) = \beta_0 + \int_{\mathcal{T}} \beta(t) X_i(t)\, dt$$

Here

- Y_i is a scalar response
- $X_i(t)$ is a functional predictor taking values in a set $\mathcal{X}$
- $\mathcal{T}$ is a compact interval
  - WLOG, $\mathcal{T} = [0, 1]$
- $\beta(t)$ is a functional parameter
- $X_i(t)$ is observed on $\{j/m : \ j = 0, \ldots, m\}$ (for simplicity)

FLM: $\beta(t)$ is nonparametric

$$E(Y_i|X_i) = \beta_0 + \int_{\mathcal{T}} \beta(t)X_i(t)\,dt \approx \beta_0 + \text{constant} \sum_{j=1}^{J} \beta(t_j)X_i(t_j)$$

The right-hand side is a high-dimensional linear model.

We expect that β is smooth but otherwise is unknown.

Therefore,

- $\beta(t)$ is modeled nonparametrically, but
- smoothness of $\beta(t)$ is imposed.
- so, nonparametric estimation is needed

Estimation must be nonparametric.

- This suggests using kernel methods.
- But kernels will not work.
- Let's see why.

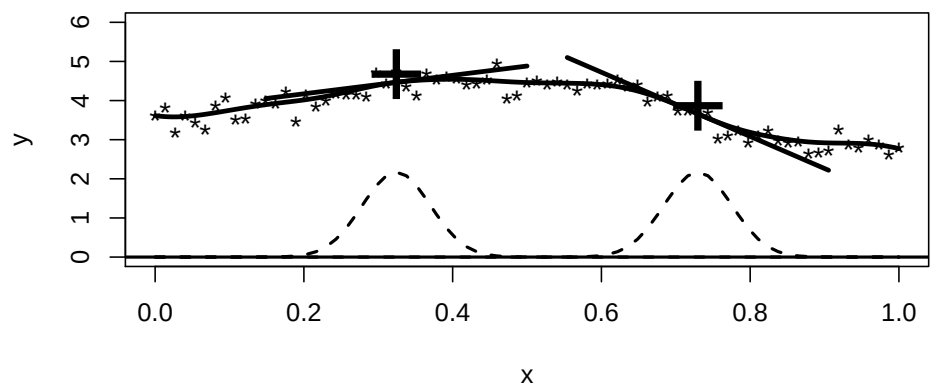
Local Kernel-Weighted Linear Regression

$$Y_i = m(X_i) + \epsilon_i \quad \text{(nonparametric regression model)}$$

$\widehat{m}(x) = \beta_0$ where (β_0, β_1) minimize

$$\sum_{i=1}^{n} K\left\{h^{-1}(X_i - x)\right\}\left[Y_i - \{\beta_0 + \beta_1(X_i - x)\}\right]^2.$$

Modeling is local and estimation is local.



In the FLM, the response depends on the entire range of X since

$$E(Y_i|X_i) = \beta_0 + \int_{\mathcal{T}} \beta(t) X_i(t)\, dt.$$

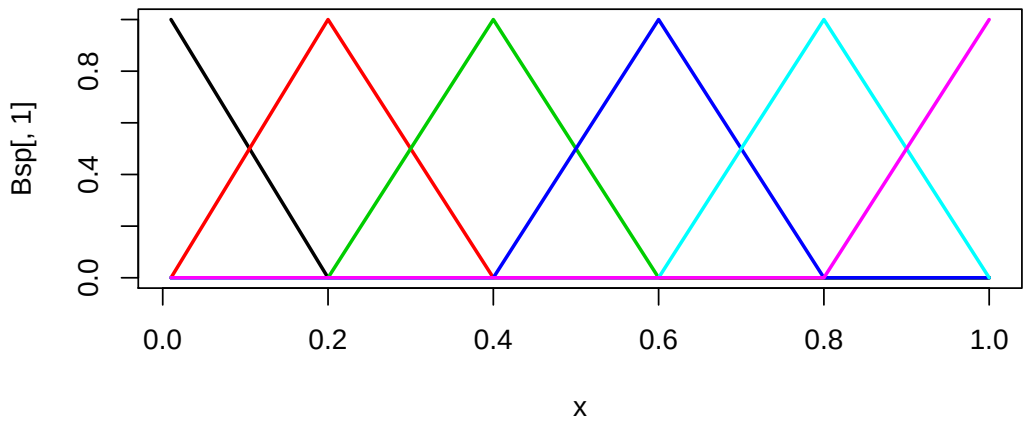
So local estimation does not work.

Modeling must still be local, since

$$\text{Local modeling} \iff \text{nonparametric.}$$

Splines: local modeling and global estimation

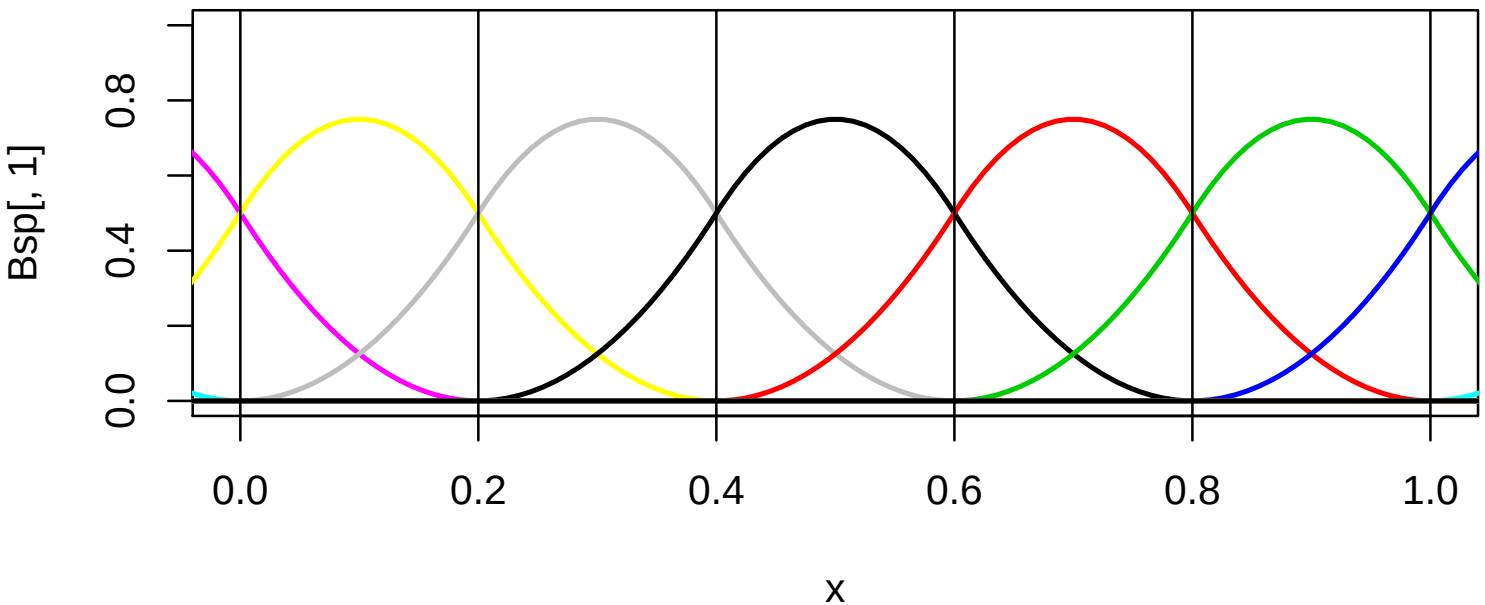
Linear B-splines:



Linear spline: $\sum_{j=1}^{J} b_j B_j(x)$. $\quad \mathbf{b} := (b_1, \ldots, b_J)$

Slopes: $b_j - b_{j-1} = (\Delta \mathbf{b})_{j-1}$ $\quad$ Changes in slopes: $\Delta^2 \mathbf{b}_{j-2}$

Quadratic splines



Degree $\geq 2 \Longrightarrow \Delta^2 \mathbf{b}_{j-2}$ is proportional to the 2nd derivative

$$E(Y_i|X_i) = \beta_0 + \int \beta(t)X_i(t)\,dt \quad \text{(regression model)}$$

$$\beta(t) = \sum_{j=1}^{J} \beta_j B_j(t) \quad \text{(spline model—local)}$$

Therefore,

$$E(Y_i|X_i) = \beta_0 + \sum_{j=1}^{J} \beta_j \underbrace{\int B_j(t)X_i(t)\,dt}_{:=U_{ij}\ \text{(known)}}$$

So estimate $\beta(t)$ by minimizing

$$\sum_{i=1}^{n} \left\{ Y_i - \sum_{j=1}^{J} \beta_j U_{ij} \right\}^2 + \lambda \sum_{j=1}^{J-2} \left\{ (\Delta^2 \mathbf{b})_j \right\}^2 \quad \text{(global estimation)}$$

Functional Generalized Additive Model

We have extended the FLM to the Functional Generalized Additive Model (FGAM)

$$g\{E(Y_i|X_i)\} = \theta_0 + \int_{\mathcal{T}} F\{X_i(t), t\}dt.$$

- $F(\cdot, \cdot)$ is an unknown function from $\mathcal{X} \times \mathcal{T}$ to $\Re$
- if $F(x, t) = x\beta(t)$, then we have a FGLM
- compare with the usual GAM with K scalar predictors

$$g\{E(Y_i|X_i)\} = \theta_0 + \sum_{k=1}^{K} F\{X_i(k), k\}$$

The FGAM

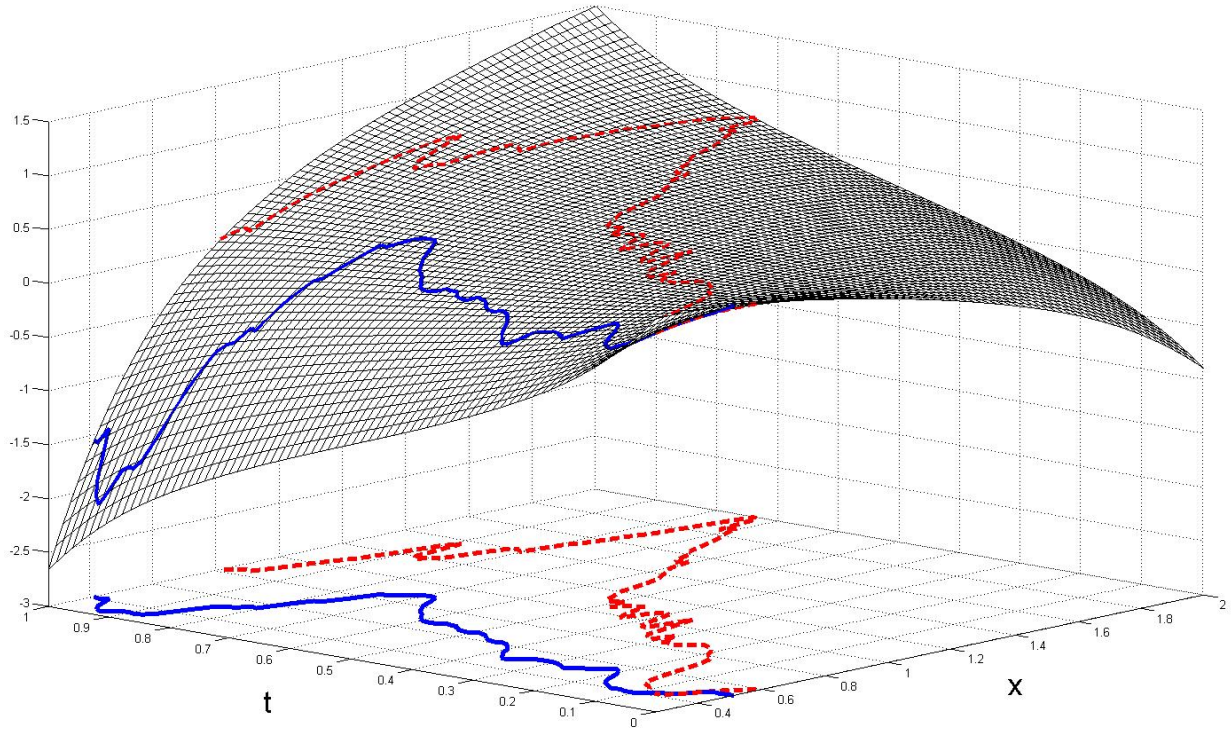
$$g\{E(Y_i|X_i)\} = \theta_0 + \int_{\mathcal{T}} F\{X_i(t), t\}dt.$$

can be approximated by a Riemann sum:

$$g\{E(Y_i|X_i)\} = \theta_0 + \Delta^{-1}\sum_j F\{X_i(t_j), t_j\}.$$

- $\{t_j : j = 1, \ldots, J\}$ is a fine grid
- Δ is the distance between grid points
- The last model is like an ordinary GAM, but
  - in the FGAM, $F(x, t)$ should be smooth in both variables

DTI example: Predicting MS using perpendicular diffusivity



Estimated surface $\hat{F}(x, t)$ and two predictor curves for the DTI (Diffusion Tensor Imaging) dataset. MS = multiple sclerosis

Advantage of penalized splines for a FGAM

Penalized splines let us impose smoothness in both variables.

Also, a penalized spline additive model can be fit in one step.

- there is no need for backfitting
  - this was first noticed by Marx and Eilers
  - this is crucial for the FGAM
  - cannot backfit with infinitely many components

We use the bivariate tensor product B-spline model:

$$F(x,t) = \sum_{j=1}^{K_x} \sum_{k=1}^{K_t} \theta_{j,k} B_j^X(x) B_k^T(t)$$

- Here $\{B_j^X(x) : j = 1, \ldots, K_x\}$ and $\{B_k^T(x) : k = 1, \ldots, K_t\}$ are univariate B-spline bases
- a roughness penalty will be imposed on $\{\theta_{j,k}\}_{j=1\,k=1}^{K_x\ K_t}$

From previous slide:

$$F(x,t) = \sum_{j=1}^{K_x} \sum_{k=1}^{K_t} \theta_{j,k} B_j^X(x) B_k^T(t)$$

Therefore,

$$g\{E(Y_i|X_i)\} = \theta_0 + \int_{\mathcal{T}} F\{X_i(t), t\} dt = \theta_0 + \sum_{j=1}^{K_x} \sum_{k=1}^{K_t} \theta_{j,k} Z_{j,k}(i)$$

- Here $Z_{j,k}(i) = \int_{\mathcal{T}} B_j^X\{X_i(t)\} B_k^T(t)\, dt$
- The integral can be approximated numerically

Identifiability constraint

For identifiability, we use the constraint

$$\sum_{j=1}^{K_x} \theta_{j,k} = 0, \;\; k = 1, \ldots, K_t$$

The row penalty is

$$\lambda_1 \sum_{j=d+1}^{K_x} (\Delta_j^d \theta_{j,k})^2$$

- $\Delta_j^d \theta_{j,k}$ is the dth difference of $\theta_{j-d,k}, \ldots, \theta_{j,k}$ (k held fixed)

The column penalty is

$$\lambda_2 \sum_{k=d+1}^{K_t} (\Delta_k^d \theta_{j,k})^2$$

- $\Delta_k^d \theta_{j,k}$ is the dth difference of $\theta_{j,k-d}, \ldots, \theta_{j,k}$ (j held fixed)

These were introduced by Marx and Eilers for bivariate P-splines.

$$\widehat{\boldsymbol{\theta}}_k = \left(\mathbb{K}^T\mathbb{Z}^T\mathbb{Z}\mathbb{K} + \lambda_1\mathbb{K}^T\mathbb{P}_1^T\mathbb{P}_1\mathbb{K} + \lambda_2\mathbb{K}^T\mathbb{P}_2^T\mathbb{P}_2\mathbb{K}\right)^{-1}\mathbb{K}^T\mathbb{Z}^T\mathbf{Y}.$$

- For simplicity, $g(x) = x$ is assumed in this talk
- $\widehat{\boldsymbol{\theta}}$ is the vector of estimated coefficients
- $\mathbb{K}$ imposes the identifiability constraints
- $\mathbb{Z}$ contains the $Z_{j,k}(i)$ values
- $\mathbb{P}_1$ and $\mathbb{P}_2$ impose the row and column penalties
- $\mathbf{Y}$ is the vector of responses

Variant of the model

The additive model is preserved by transforming $X_i(t)$ to $G_t\{X_i(t)\}$ where $G_t(x)$ is "smooth" in both t and x.

- We use G_t equal to the CDF of $X_i(t)$
  - We use the empirical EDF (perhaps smoothed)
  - Then the set $\left[j/m, G_{j/m}\{X_i(j/m)\}\right]_{i=1}^{n}{}_{j=0}^{m}$ fills $[0,1]^2$
  - Convenient both for visualization and estimation
  - As nice interpretation: $F(p,t)$ is the effect of $X_i(t)$ when at its pth quantile

Variant of the model, cont.

- Transforming $X_i(t)$ is a reexpression of the original model, not a new model
  - that would not be true if $X_i(t)$ were transformed in a FLM

Smoothing parameters and inference

- we select the smoothing parameters λ_1 and λ_2 by GCV
  - can be computed rapidly using software of Simon Wood in his `mgcv` package in `R`
  - we used "outer iteration" where, for each pair of smoothing pairs, P-IRLS is applied until convergence
  - our code will be available as a supplement to the paper
  - Matt McLean intends to make an FGAM function available in the `refund` package (Ciprian Crainiceanu and Philip Reiss coordinating authors)
- the estimate is linear in $\mathbf{Y}$, if we ignore that λ_1 and λ_2 are data-based
  - standard errors obtained by a sandwich formula

Monte Carlo: Data Generation

We used the data generation scheme of Hall and Horowitz.

$X_i(t)$ was a mean-zero process with

- eigenvalues of the covariance matrix either "well spaced" or "closely spaced" and
- decay of eigenvalues determined by $\alpha = 1.2$ or 2.

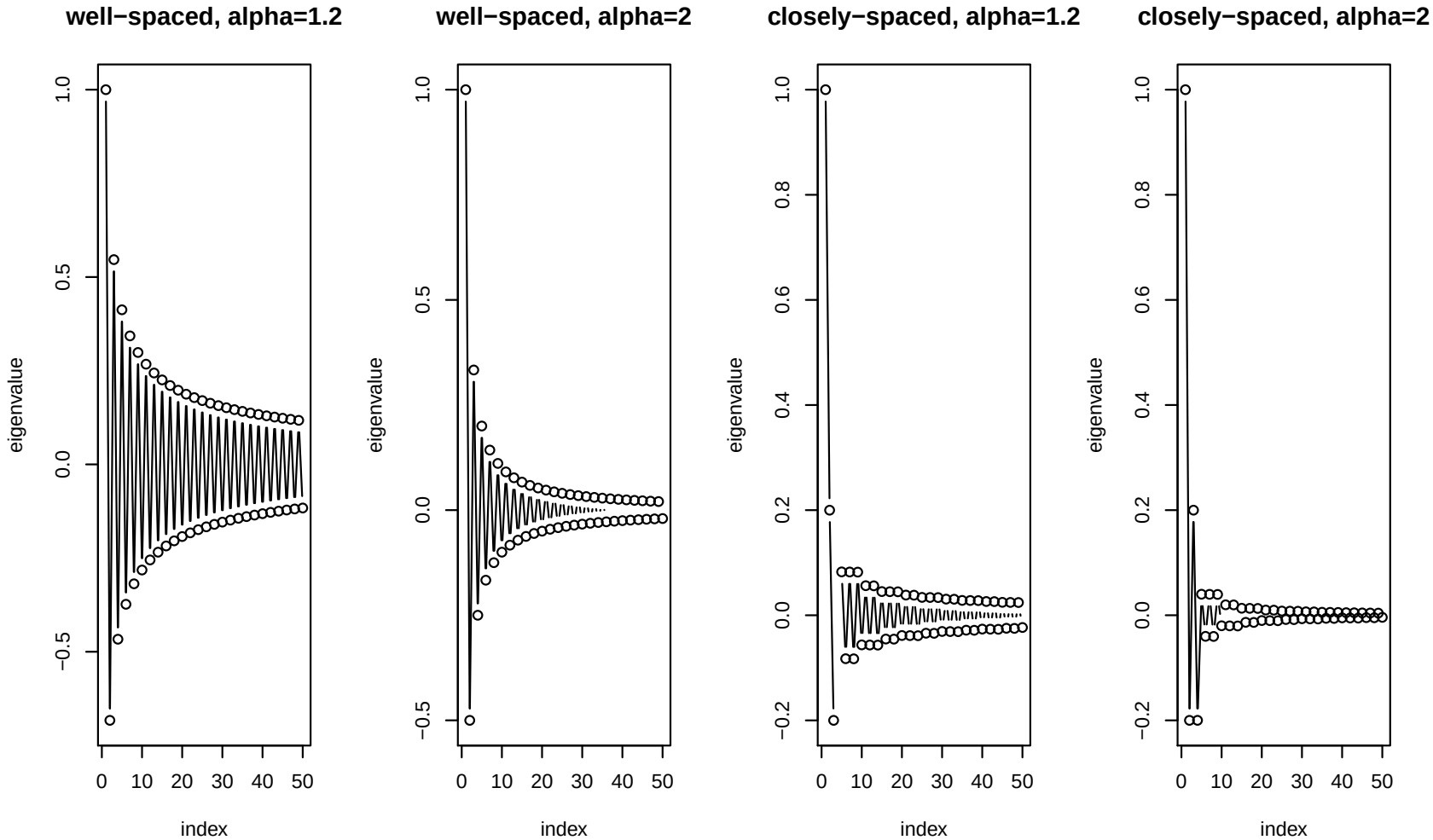
The regression model was either a FLM or a nonlinear FGAM.

Generating the X_i

$$X_i(t) = \sum_{j=1}^{50} \gamma_j Z_{ij} \phi_j(t)$$

- $\gamma_1, \ldots, \gamma_{50}$ are the eigenvalues of the covariance operator
- Z_{ij} are iid uniform$(-3.5, 3.5)$
- $\phi_1(t), \ldots, \phi_{50}(t)$ are the eigenvectors (eigenfunctions) of the covariance operator

Generating the X_i: Eigenvalues

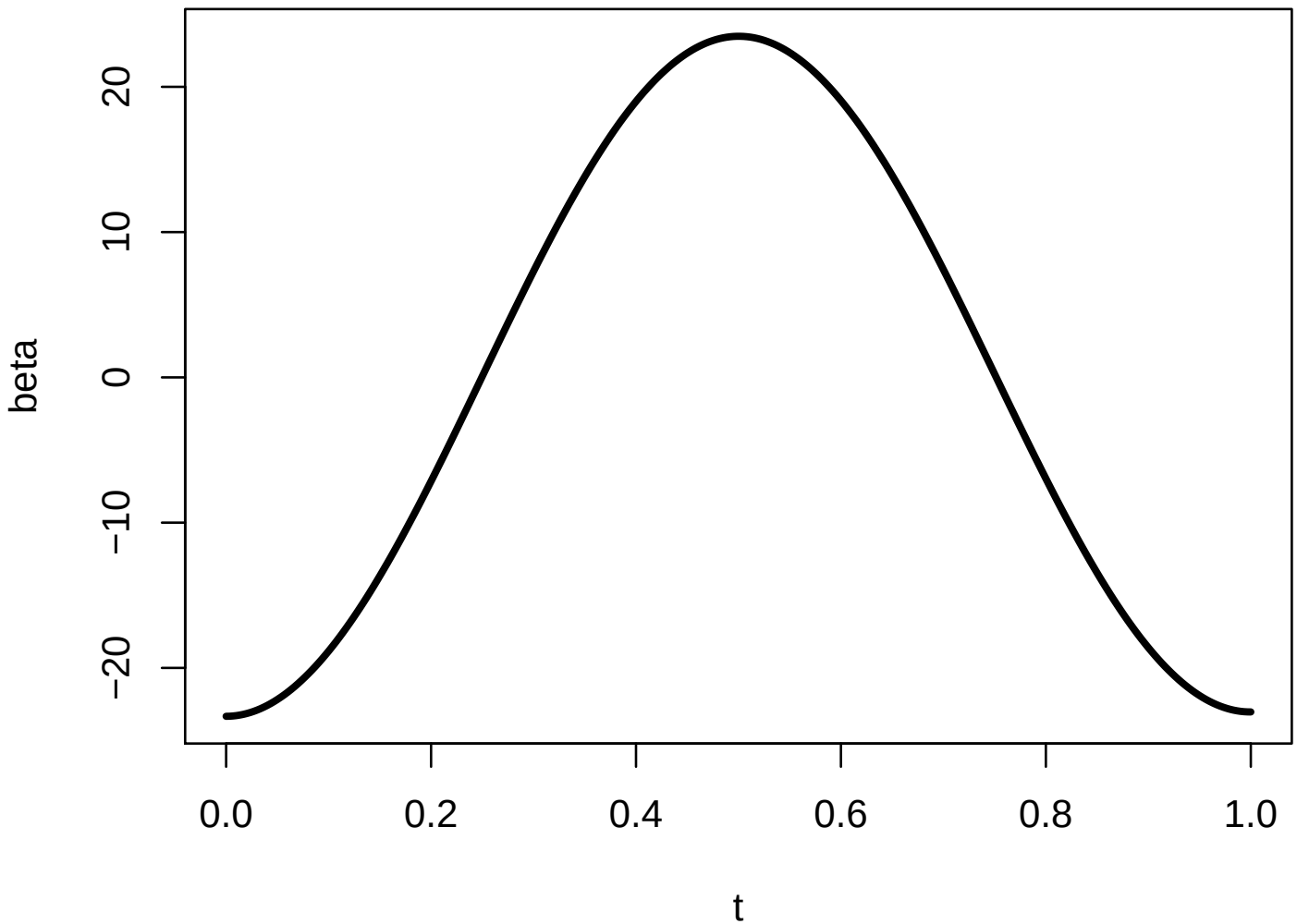


Generating the X_i: Eigenvectors

$$\phi_1(t) \equiv 1$$

$$\phi_j(t) = \sqrt{2}\cos(j\pi t),\ j = 2, \ldots, 50$$

Generating the Y_i: True Coefficient Function



Generating the Y_i: True Models

Linear

$$Y_t = \int_0^1 \beta(t) X_i(t)\, dt + \epsilon_i$$

Additive

$$Y_t = \int_0^1 \{\beta(t) X_i(t)\}^2\, dt + \epsilon_i$$

Five Estimators

1. FGAM-O: FGAM with Original X_i (no transformation)
2. FLM1: Functional Linear Model using `fRegress` in the `fda` package
3. FLM2: Functional Linear Model using `pca.fd` in the `fda` package
   - an unpenalized OLS fit to PCA scores. Enough eigenvectors are used to explain 90% of the variability in the X_i
4. FV: Ferraty-Vieu kernel estimation $=$ Nadaraya-Watson type estimator
5. FAM: due to Müller and Yao (2008)
   - an additive model fit to a finite set of PCA scores

FLM1 and FLM2 are similar

Let $\mathbf{x}_i$ and $\boldsymbol{\beta}$ be the evaluations of $X_i(t)$ and $\beta(t)$ on fine grids. Then

$$\int X_i(t)\beta(t)dt \approx \text{constant } \mathbf{x}_i^T\boldsymbol{\beta} = (\mathbf{A}\mathbf{x}_i)^\mathsf{T}(\mathbf{A}^{-T}\boldsymbol{\beta})$$

so a FLM is invariant to linear transformations of the predictor.

So regressing on $X_i(t)$ is, in principle, equivalent to regressing on PCA scores.

Therefore, FLM1 and FLM2 are using the same model

- they differ only the details of their implementations.

Müller and Yao's FAM

- Start with function PCA
- Then extract a finite number of scores (projections onto eigenvectors).
- Next use an additive model in scores.
- So Müller and Yao's model is

$$Y_i = \theta_0 + \sum_{k=1}^{K} F_k \left\{ \int \xi_k(t) X_i(t)\, dt \right\} \approx \theta_0 + \sum_{k=1}^{K} F_k \left\{ \xi_k^T X_i \right\}.$$

- $\xi_1, \ldots, \xi_K$ are the first k principal component eigenvectors.
- this is a completely different model than FGAM, not just a different method of estimation.
  - additive in $X_i(t) \Rightarrow$ not additive in scores (and vice versa).
  - see next frame.

Additive Models and Linear Transformations

Additive models are not invariant to linear transformation.

To appreciate this, consider a bivariate additive model

$$f_1(X_1) + f_2(X_2).$$

Consider the transformations $Z_1 = \alpha_{11}X_1 + \alpha_{12}X_2$ and $Z_2 = \alpha_{21}X_1 + \alpha_{22}X_2$.

If f_1 and f_2 are nonlinear, then in general there will not exist g_1 and g_2 such that

$$f_1(X_1) + f_2(X_2) = g_1(Z_1) + g_2(Z_2).$$

Therefore, a model that is additive in the original X_t will not be additive in the PCA scores, and vice versa.

It follows that FGAM (additive in X_t) and FAM (additive in the PCA scores) are fundamentally different models.

Exception: FGAM and FAM are the same if applied to a linear model.

$$\widehat{r}(X) = \frac{\sum_{i=1}^{N} Y_i \, K\left\{h^{-1} d(X, X_i)\right\}}{\sum_{i=1}^{N} K\left\{h^{-1} d(X, X_i)\right\}}.$$

- $\widehat{r}(X)$ estimates $E(Y_i | X_i = X)$.
- $d(\cdot, \cdot)$ is a semimetric on the space of functions under consideration.
- h is a bandwidth.
- K is a kernel.
- This estimator is a generalization of the Nadaraya-Watson kernel estimator.
- Uses code on the authors' website.

Mean out-of-sample RMSEs

			$Y_i = \int_{\mathcal{T}} \beta(t)X_i(t)dt + \epsilon_i$					$Y_i = \int_{\mathcal{T}} \{\beta(t)X_i(t)\}^2 dt + \epsilon_i$				
γ_j^2	σ_ϵ	α	FGAM-O	FLM1	FLM2	FV	FAM	FGAM-O	FLM1	FLM2	FV	FAM
well spaced	0.5	1.1	0.52	0.52	0.61	0.75	0.82	0.69	8.28	5.75	4.27	6.98
		2.0	0.52	0.52	0.52	0.59	0.55	0.64	3.87	2.75	1.80	2.49
	1.0	1.1	1.04	1.03	1.21	1.18	1.65	1.27	8.29	5.85	4.38	7.22
		2.0	1.03	1.02	1.04	1.08	1.09	1.19	4.03	2.95	2.07	2.72
closely spaced	0.5	1.1	0.52	0.51	0.52	0.55	0.53	0.60	2.62	1.82	1.11	1.32
		2.0	0.52	0.51	0.54	0.55	0.55	0.58	2.48	1.74	0.98	1.20
	1.0	1.1	1.03	1.03	1.03	1.06	1.07	1.13	2.75	2.01	1.45	1.61
		2.0	1.03	1.03	1.06	1.06	1.05	1.14	2.65	1.97	1.37	1.53

Diffusion Tensor Imaging

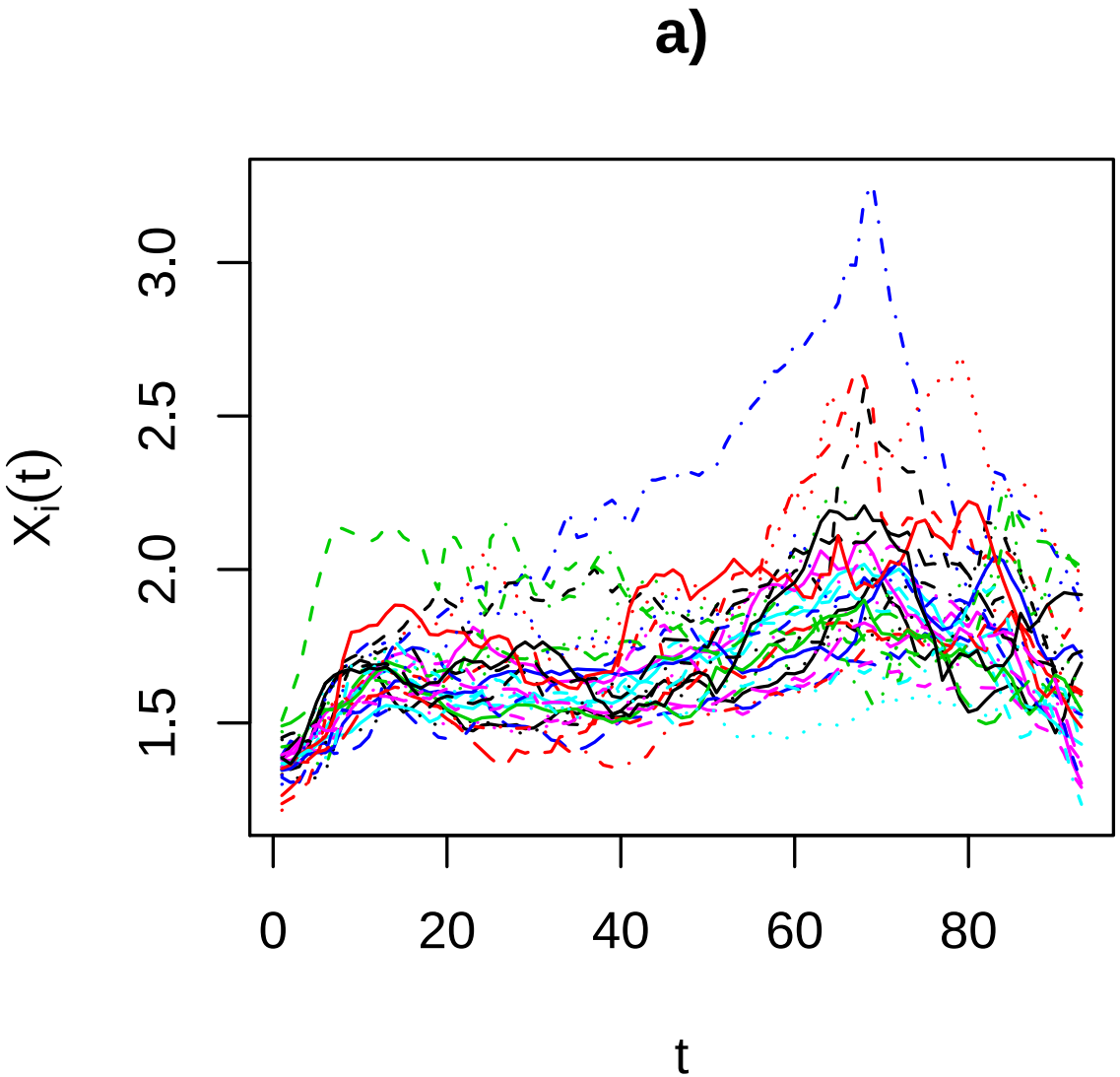
- DTI = diffusion tensor imaging
- Data is courtesy of Dr. Daniel Reich, National Institute of Neurological Disorders and Stroke and Johns Hopkins University, Department of Neurology
- at each voxel, get a 6-dimensional tensor = components of a 3×3 symmetric, PD matrix
    - measures diffusion of water

Multiple Sclerosis

- a “tract” is white matter connecting two parts of the brain
- MS is a disease of the white matter
  - specifically of the white myelin sheave
  - the myelin is an insulator
- the corpus callosum tract connects the two hemispheres of the brain
  - involved in many brain functions including cognition
- this is an exploratory study
  - Question: how is diffusion along the corpus callosum related to cognition?

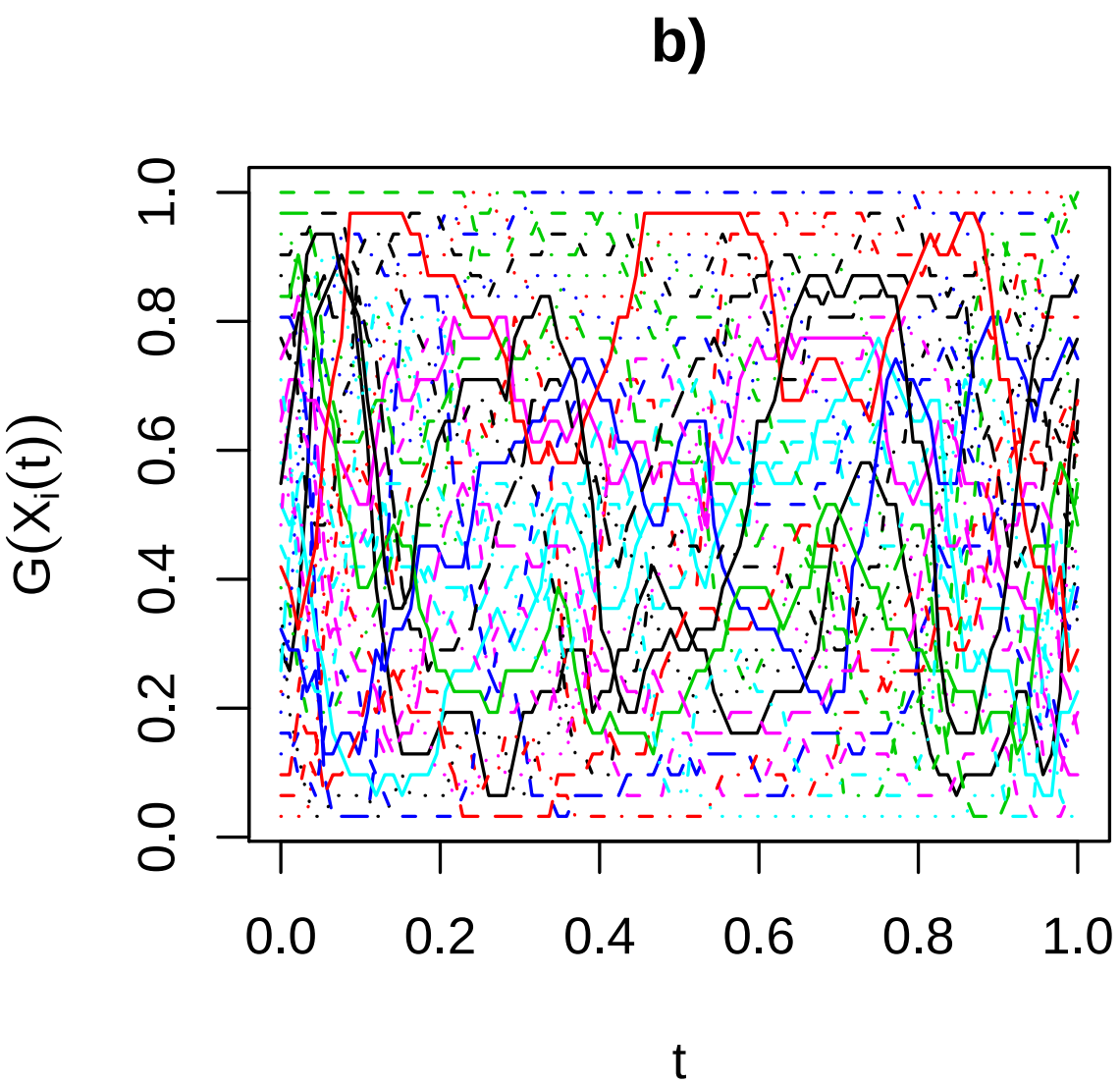
DTI data – parallel diffusivity along the corpus callosum

MS patients only – Untransformed



DTI data – parallel diffusivity along the corpus callosum

MS patients only – Transformed



PASAT

- PASAT = Paced Auditory Serial Addition Test
- Subject given numbers at three second intervals
  - asked to add the current number to the previous one
- MS patients often perform significantly worse than controls
  - We only have data from patients

Predicting PASAT: Leave-one-curve-out RMSEs

Measurement	FGAM-O	FGAM-T	FLM1	FLM2	FV	FAM
Perp. Diff.	12.22	10.46	10.98	11.27	11.16	11.71
Fract. Aniso.	12.55	11.60	11.87	11.91	12.11	12.70
Parall. Diff.	11.94	12.09	12.32	12.24	11.97	11.86

RMSE = Root Mean Squared Error

FGAM-O = FGAM with original curves

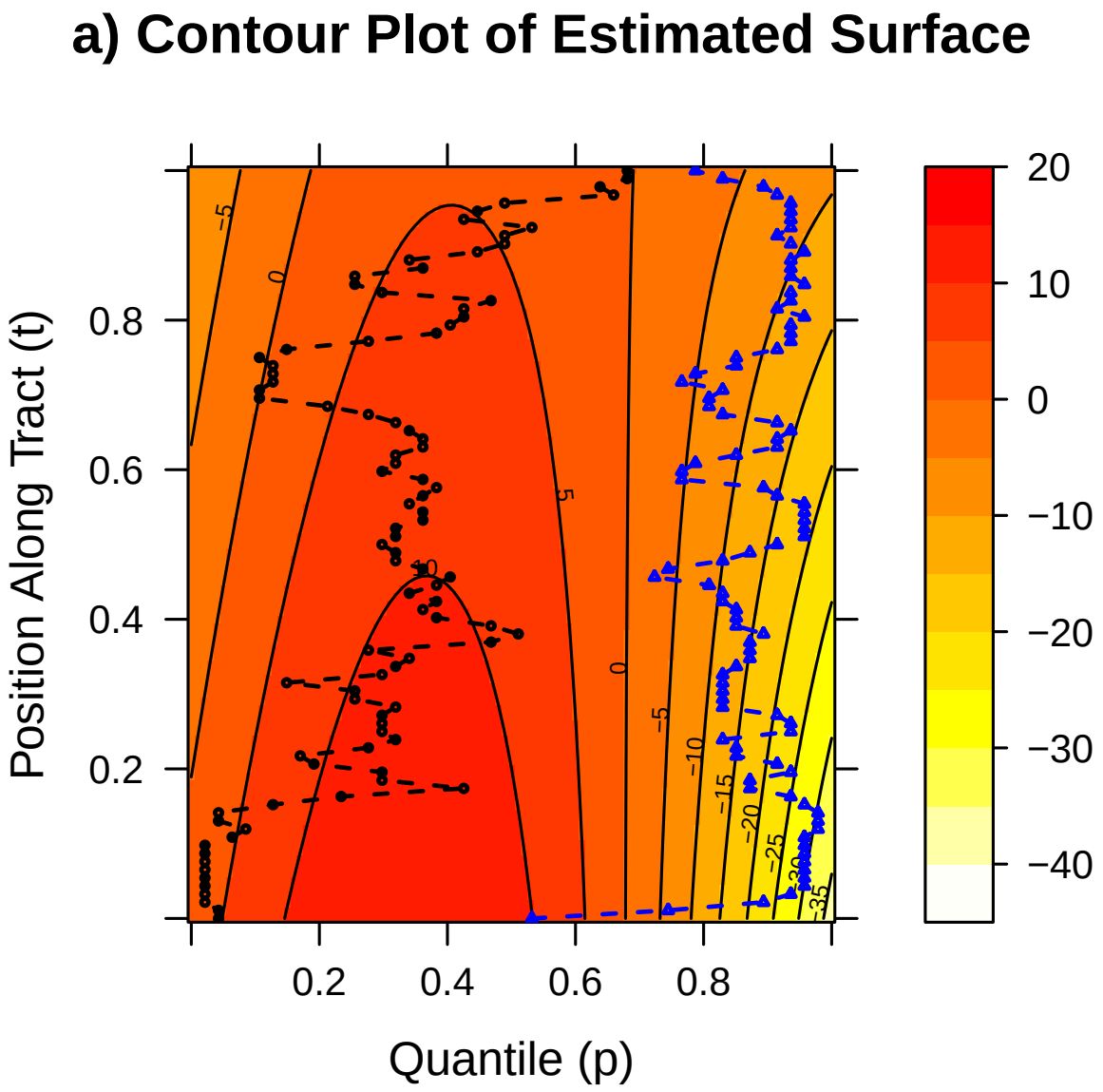
FGAM-T = FGAM with ECDF transformed curves

Perp. Diff. = Perpendicular Diffusivity

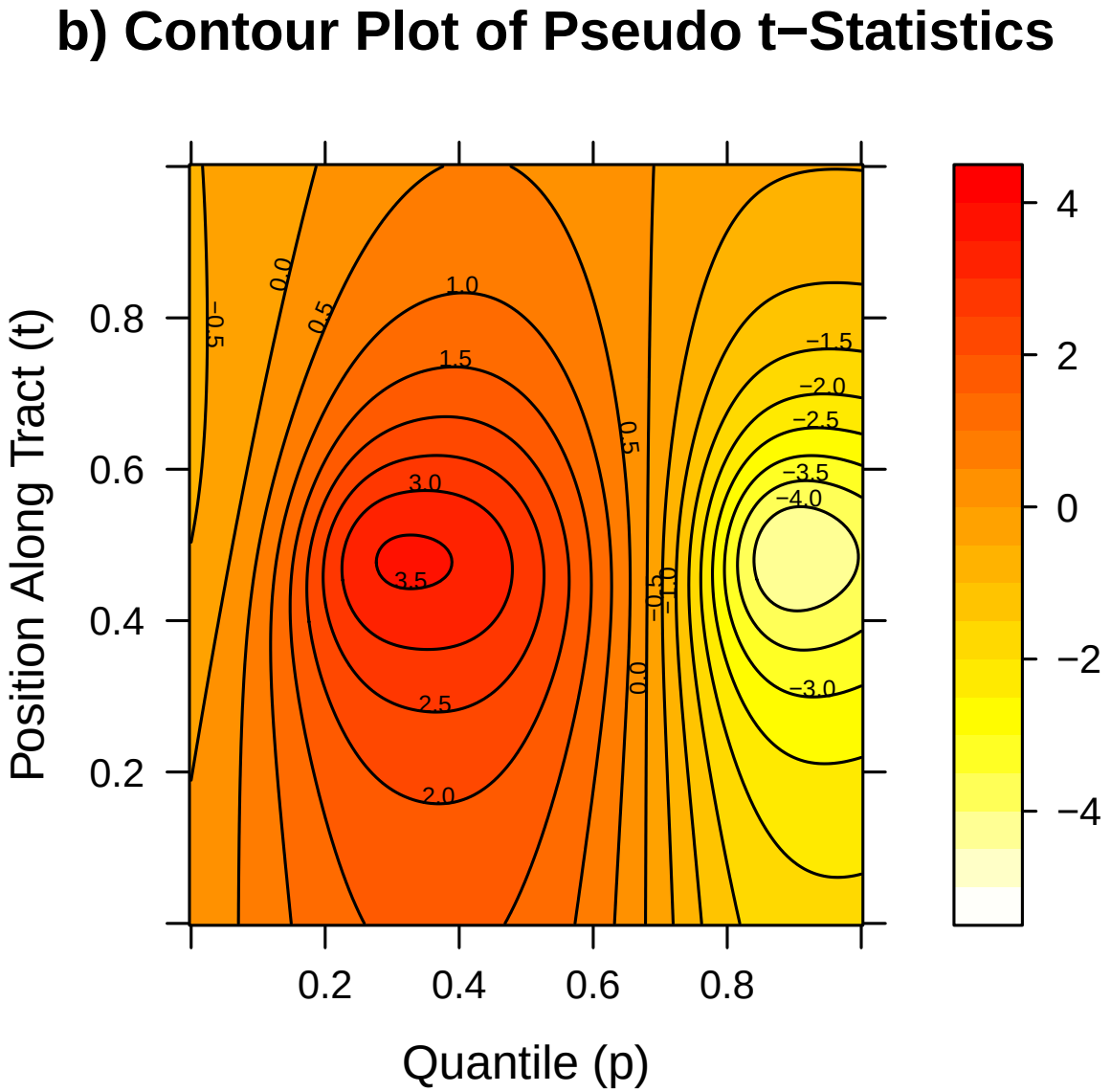
Fract. Aniso. = Fractional Anisotropy

Parall. Diff. = Parallel Diffusivity

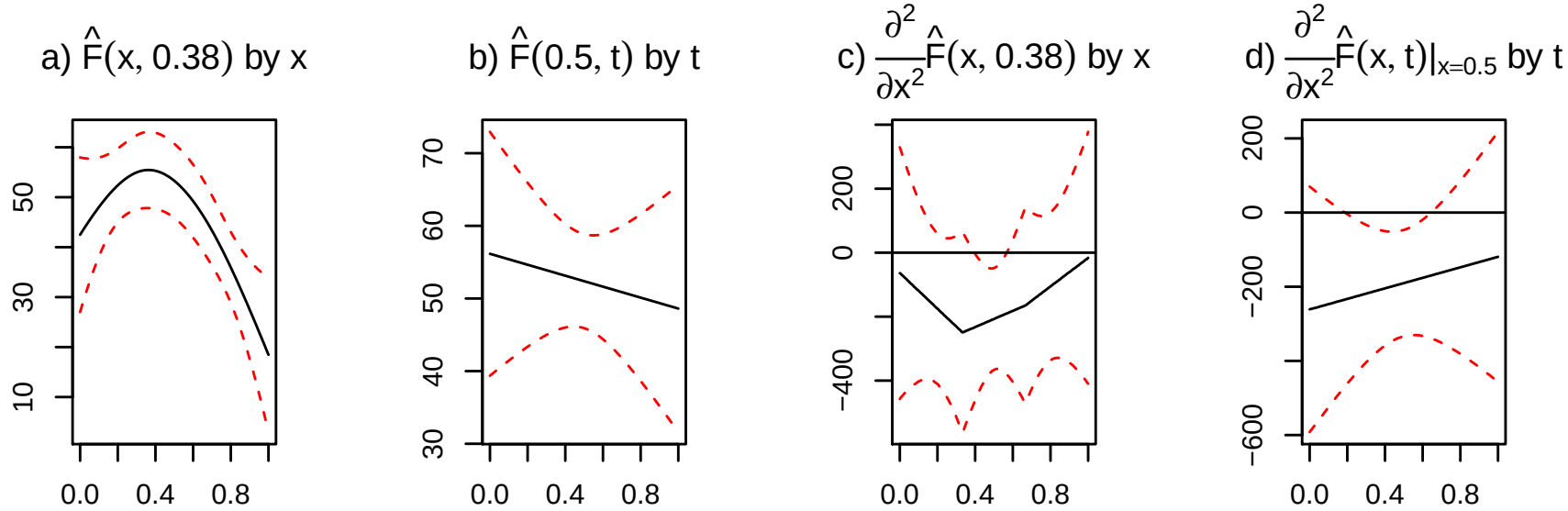
Estimated surface $\widehat{F}(p, t)$ for parallel diffusivity and PASAT



t-statistics for parallel diffusivity and PASAT



Slice of the fitted surface (parallel diffusivity and PASAT)



For an FLM $F(x, t) = x\beta(t)$.

A non-zero value of $\frac{\partial^2}{\partial x^2}\widehat{F}(x, t)$ indicates that an FLM does not fit well.

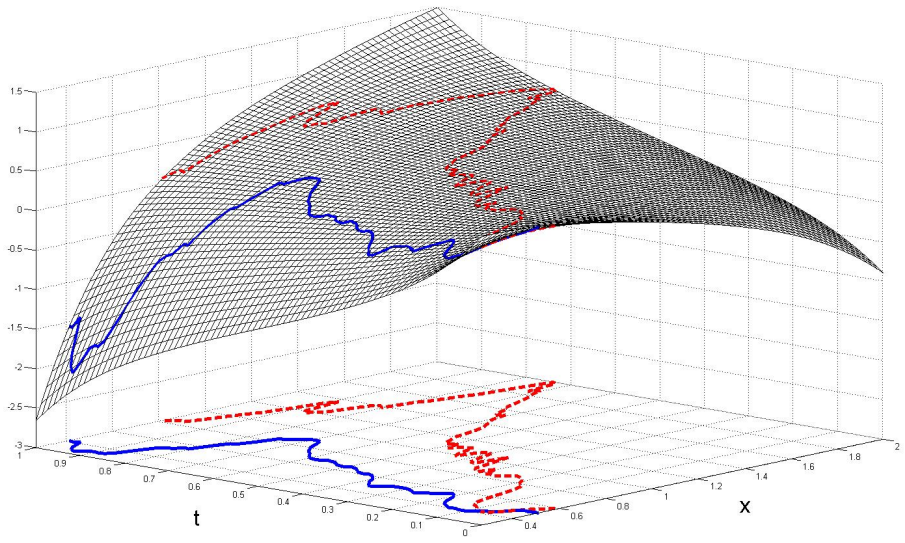
DTI data: Predicting MS status

Now the response is binary: Multiple sclerosis patient/control.

The goal isn't really classification, since there are better ways to diagnose the disease.

Instead, we want to see what can be learned about the disease.

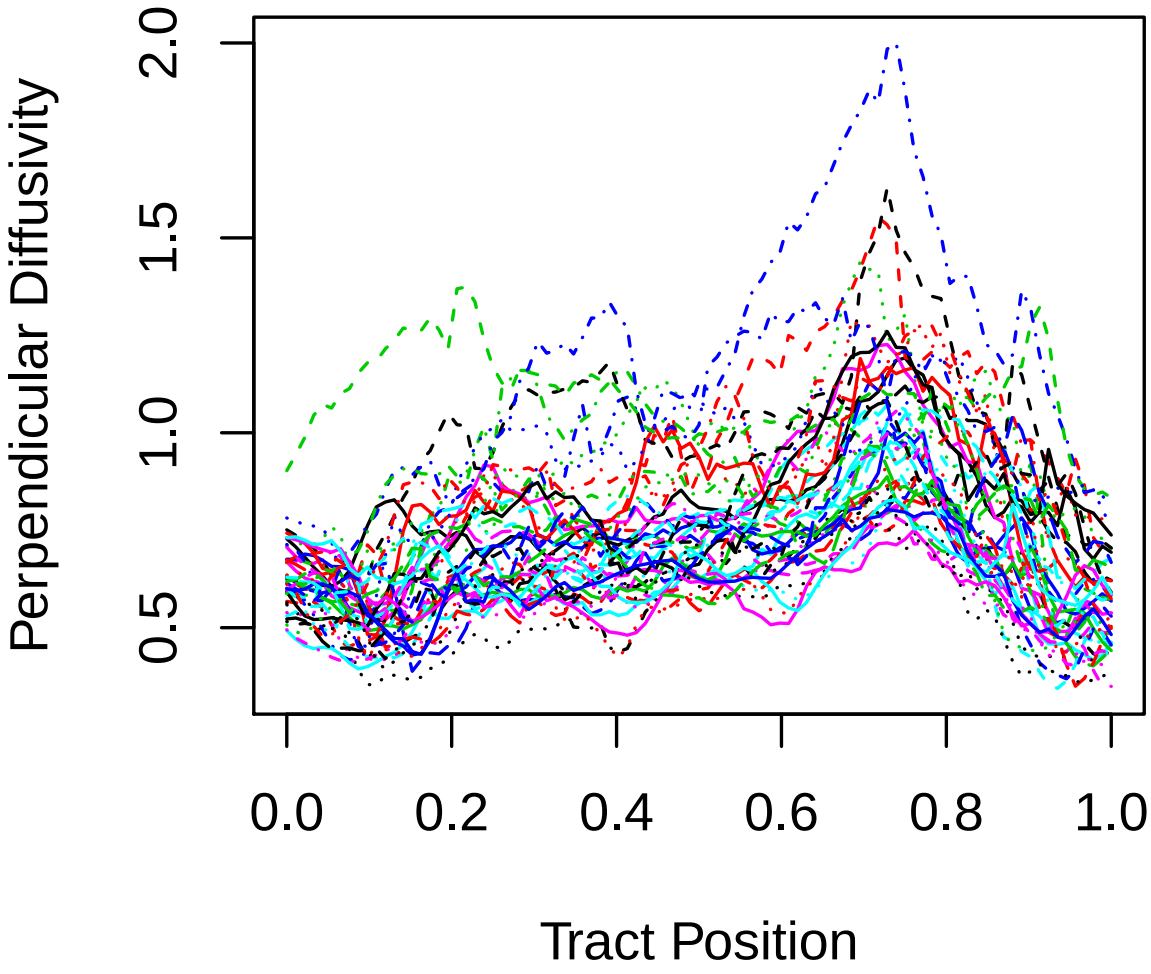
DTI example: Predicting MS using perpendicular diffusivity



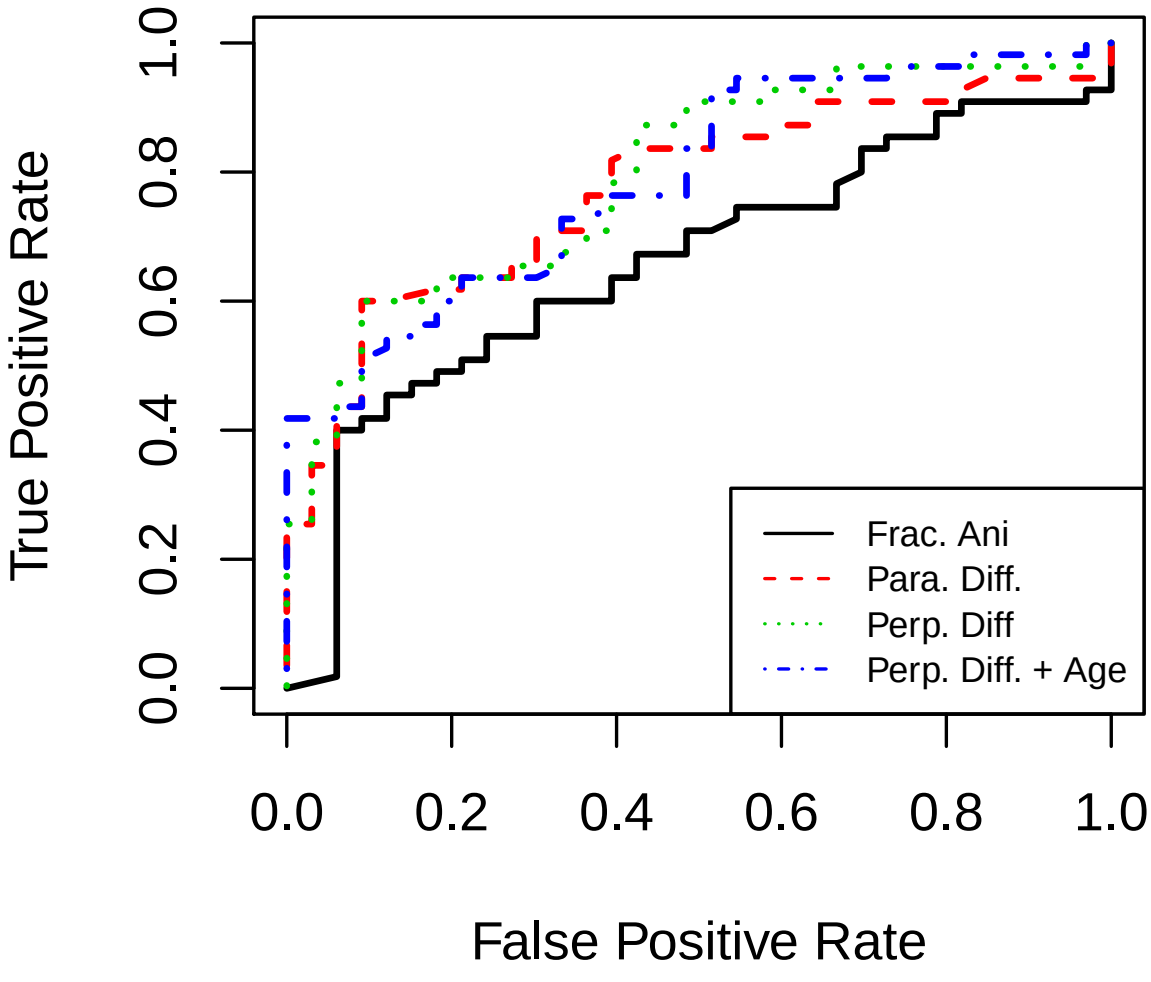
Estimated surface $\hat{F}(x,t)$ and two predictor curves for the DTI (Diffusion Tensor Imaging) dataset. Quantile transformation used.

Note that large quantiles at tract locations near $t = 1$ have a strong influence on predicting disease.

Perpendicular Diffusivity Curves Along Corpus Callosum



ROC curves


True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
Frac. Ani
Para. Diff.
Perp. Diff
Perp. Diff. + Age