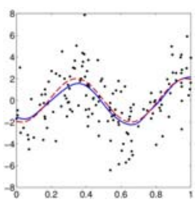
School of Computer Science
Carnegie Mellon

Probabilistic Graphical Models

Structured Sparse Additive Models



Junming Yin and Eric Xing

Lecture 27, April 24, 2013

Reading: See class website

1

Outline

- Nonparametric regression and kernel smoothing
- Additive models
- Sparse additive models (SpAM)
- Structured sparse additive models (GroupSpAM)

2

Nonparametric Regression and Kernel Smoothing

Non-linear functions:

LR with non-linear basis functions

- LR does not mean we can only deal with linear relationships
- We are free to design (non-linear) features under LR

$$y = \theta_0 + \sum_{j=1}^{m} \theta_j \phi(x) = \theta^T \phi(x)$$

where the $\phi_j(x)$ are fixed basis functions (and we define $\phi_0(x) = 1$).

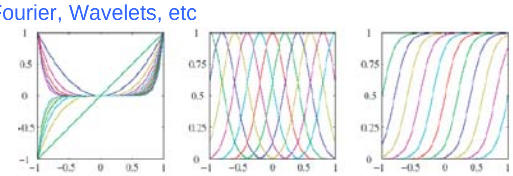
- Example: polynomial regression:

$$\phi(x) := \left[1, x, x^2, x^3\right]$$

- We will be concerned with estimating (distributions over) the weights θ and choosing the model order M.

Basis functions

- There are many basis functions, e.g.:
  - Polynomial $\phi_j(x) = x^{j-1}$
  - Radial basis functions $\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$
  - Sigmoidal $\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$
  - Splines, Fourier, Wavelets, etc

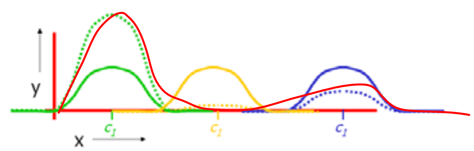


1D and 2D RBFs

- 1D RBF

$$y^{est} = \beta_1 \phi_1(x) + \beta_2 \phi_2(x) + \beta_3 \phi_3(x)$$

- After fit:

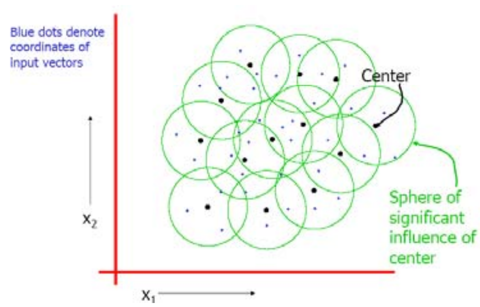


$$y^{est} = 2\phi_1(x) + 0.05\phi_2(x) + 0.5\phi_3(x)$$

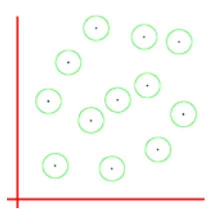
7

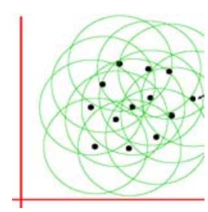
Good and Bad RBFs

- A good 2D RBF



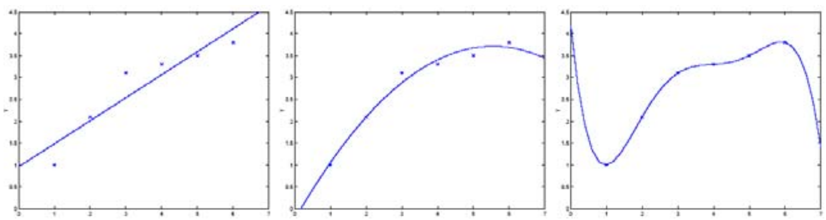
- Two bad 2D RBFs





8

Overfitting and underfitting

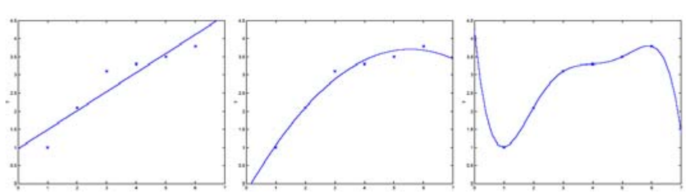


$y=\theta_0+\theta_1 x$ $\quad$ $y=\theta_0+\theta_1 x+\theta_2 x^2$ $\quad$ $y=\sum_{j=0}^{5}\theta_j x^j$

9

Bias and variance

- We define the bias of a model to be the expected generalization error even if we were to fit it to a very (say, infinitely) large training set.
- By fitting "spurious" patterns in the training set, we might again obtain a model with large generalization error. In this case, we say the model has large variance.



10

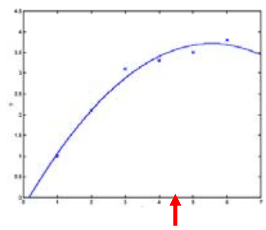
Locally weighted linear regression

- The algorithm:

  Instead of minimizing $J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^T\theta - y_i)^2$

  now we fit θ to minimize $J(\theta) = \frac{1}{2}\sum_{i=1}^{n} w_i(\mathbf{x}_i^T\theta - y_i)^2$

  Where do w_i's come from? $w_i = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x})^2}{2\tau^2}\right)$



  - where $\mathbf{x}$ is the query point for which we'd like to know its corresponding $\mathbf{y}$

→ Essentially we put higher weights on (errors on) training examples that are close to the query point (than those that are further away from the query)

Parametric vs. non-parametric

- Locally weighted linear regression is another example we are running into of a **non-parametric** algorithm. (what are the others?)
- The (unweighted) linear regression algorithm that we saw earlier is known as a **parametric** learning algorithm
  - because it has a fixed, finite number of parameters (the θ), which are fit to the data;
  - Once we've fit the θ and stored them away, we no longer need to keep the training data around to make future predictions.
  - In contrast, to make predictions using locally weighted linear regression, we need to keep the entire training set around.
- The term "non-parametric" (roughly) refers to the fact that the amount of stuff we need to keep in order to represent the hypothesis grows linearly with the size of the training set.

Parametric vs. non-parametric

- Parametric model:
  - Assumes all data can be represented using a fixed, finite number of parameters.
  - Examples: polynomial regression
- Nonparametric model:
  - Number of parameters can grow with sample size.
  - Examples: nonparametric regression

13

Robust regression—probabilistic interpretation

- **What regular regression does:**

  Assume y_k was originally generated using the following recipe:

$$y_k = \theta^T \mathbf{x}_k + \mathcal{N}(0, \sigma^2)$$

  Computational task is to find the Maximum Likelihood estimation of θ

14

Nonparametric Regression: Formal Definition

- Nonparametric regression is concerned with estimating the regression function

$$m(\mathbf{x}) = \mathbb{E}(Y \mid X = \mathbf{x})$$

from a training set $\{(\mathbf{x}^{(i)}, y^{(i)}) : \mathbf{x}^{(i)} \in \mathbb{R}^p, y^{(i)} \in \mathbb{R}, i = 1, \ldots, n\}$

- The “parameter” to be estimated is the whole function m(x)
- No parametric assumption such as linearity is made about the regression function m(x)
  - More flexible than parametric model
  - However, usually require keeping the entire training set (memory-based)

15

Kernel Smoother

- The simplest nonparametric regression estimator
  - Local weighted (smooth) average of $y^{(i)}$
  - The weight depends on the distance to $\mathbf{x}^{(i)}$
- Nadaraya-Watson kernel estimator

$$\hat{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} y^{(i)} K\left(\frac{\|\mathbf{x}-\mathbf{x}^{(i)}\|}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{\|\mathbf{x}-\mathbf{x}^{(i)}\|}{h}\right)}$$

  - K is the smoothing kernel function K(x)>=0 and h is the bandwidth

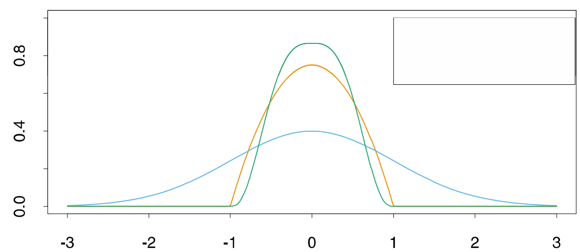
16

Kernel Function

- It satisfies

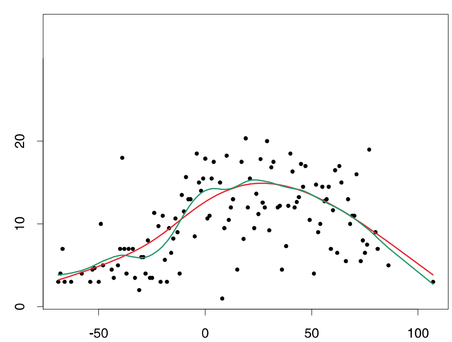
$$\int K(x)\,dx = 1, \quad \int xK(x)dx = 0 \quad \text{and} \quad \sigma_K^2 \equiv \int x^2 K(x)dx > 0.$$

- Different types



Bandwidth

- **The choice of bandwidth h is much more important than the type of kernel K**
  - **Small h -> rough estimates**
  - **Large h -> smoother estimates**
  - **In practice: cross-validation or plug-in methods**



Linear Smoothers

- Kernel smoothers are examples of linear smoothers

$$\widehat{m}(\mathbf{x}) = \sum_{i=1}^{n} \ell_i(\mathbf{x}) y^{(i)} = \ell(\mathbf{x})^T \mathbf{y},$$

$$\ell_i(\mathbf{x}) = \frac{K\left(\frac{\|\mathbf{x}-\mathbf{x^{(i)}}\|}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{\|\mathbf{x}-\mathbf{x^{(i)}}\|}{h}\right)}$$

  - For each x, the estimator is a linear combination of $y^{(i)}$
  - Other examples: smoothing splines, locally weighted polynomial, etc

19

Linear Smoothers (con't)

- **Define $\widehat{\mathbf{y}} = (\widehat{m}(\mathbf{x}^{(1)}), \ldots, \widehat{m}(\mathbf{x}^{(n)}))$ be the fitted values of the training examples, then**

$$\hat{\mathbf{y}} = \mathbf{S}\mathbf{y},$$

  - **The n x n matrix S is called the smoother matrix with $\mathbf{S}_{ij} = \ell_j(\mathbf{x}^{(i)})$**
  - **The fitted values are the smoother version of original values**

- **Recall the regression function $m(X) = \mathbb{E}(Y \mid X)$ can be viewed as**

$$m(X) = PY$$

  - **P is the conditional expectation operator $\mathbb{E}(\cdot \mid X)$ that projects a random variable (it is Y here) onto the linear space of X**
  - **It plays the role of smoother in the population setting**

20

Additive Models

Additive Models

- Due to curse of dimensionality, smoothers break down in high dimensional setting
- Hastie & Tibshirani (1990) proposed the additive model

$$m(X_1, \ldots, X_p) = \alpha + \sum_{j=1}^{p} f_j(X_j)$$

  - Each f_j is a smooth one-dimensional component function
- However, the model is not identifiable
  - Can add a constant to one component function and subtract the same constant from another component
  - Can be easily fixed by assuming

$$\mathbb{E}[f_j(X_j)] = 0 \text{ for each } j$$

Backfitting

- The optimization problem in the population setting is

$$\frac{1}{2}\mathbb{E}\left[\left(Y - \alpha - \sum_{j=1}^{p} f_j(X_j)\right)^2\right]$$

- It can be shown that the optimum is achieved at

$$\alpha = \mathbb{E}(Y), f_j = \mathbb{E}\left[\left(Y - \alpha - \sum_{k \neq j} f_k\right) \mid X_j\right] := P_j R_j$$

  - $P_j = \mathbb{E}[\cdot \mid X_j]$ is the conditional expectation operator onto jth input space
  - $R_j = Y - \alpha - \sum_{k \neq j} f_k$ is the partial residual

23

Backfitting (con't)

- Replace conditional operator P_j by smoother matrix $\mathbf{S}_j$ results in the backfitting algorithm
  - Initialize: $\hat{\alpha} = \sum_{i=1}^{n} y^{(i)}/n, \hat{\mathbf{f}}_j = 0, j = 1, \ldots, p$
  - Cycle: for $j = 1, \ldots, p, 1, \ldots, p, \ldots$

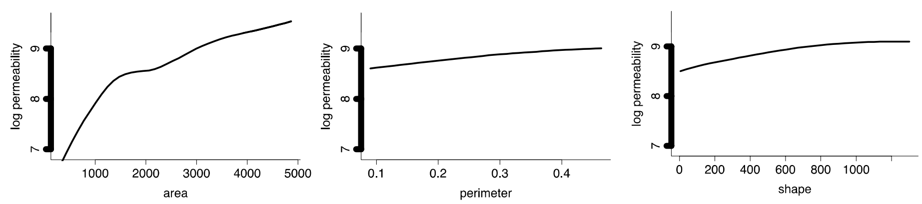
$$\hat{\mathbf{f}}_j \leftarrow \mathbf{S}_j\left(\mathbf{y} - \hat{\alpha} - \sum_{k \neq j} \hat{\mathbf{f}}_k\right), \hat{\mathbf{f}}_j \leftarrow \hat{\mathbf{f}}_j - \frac{1}{n}\sum_{i=1}^{n} \hat{f}_j(x_{ij})$$

  - $\hat{\mathbf{f}}_j = (\hat{f}_j(x_{1j}), \ldots, \hat{f}_j(x_{nj}))^T$ is the current fitted values of jth component f_j on the n training examples
  - It is a coordinate descent algorithm

24

Example

- 48 rock samples from a petroleum reservoir
- The response: permeability
- The covariates: the area of pores, perimeter in pixels and shape (perimeter/sqrt(area))



25

Sparse Additive Models (SpAM)

26

SpAM

- A sparse version of additive models (Ravikumar et. al 2009)
- Can perform component/variable selection for additive models even when n << p
- The optimization problem in the population setting is

$$\frac{1}{2}\mathbb{E}\left[\left(Y - \sum_{j=1}^{p} f_j(X_j)\right)^2\right] + \lambda \sum_{j=1}^{p} \sqrt{\mathbb{E}[f_j(X_j)^2]}$$

  - $\sum_{j=1}^{p} \sqrt{\mathbb{E}[f_j(X_j)^2]}$ behaves like an l1 ball across different components to encourage functional sparsity
  - If each component function $f_j(X_j)$ is constrained to have the linear form, the formulation reduces to standard lasso (Tibshirani 1996)

27

SpAM Backfitting

- The optimum is achieved by soft-thresholding step

$$f_j = \left[1 - \frac{\lambda}{\sqrt{\mathbb{E}[(P_j R_j)^2]}}\right]_+ P_j R_j, j = 1, \ldots, p$$

  - $R_j = Y - \sum_{k \neq j} f_k$ is the partial residual; $[\cdot]_+$ is the positive part
  - $f_j = 0$ if and only if $\sqrt{\mathbb{E}[(P_j R_j)^2]} \leq \lambda$ (thresholding condition)
- As in standard additive models, replace P_j by $\mathbf{S}_j$

$$\hat{\mathbf{f}}_j \leftarrow \left[1 - \frac{\lambda}{\hat{s}_j}\right]_+ \mathbf{S}_j \left(\mathbf{y} - \sum_{k \neq j} \hat{\mathbf{f}}_k\right), j = 1, \ldots, p$$

  - $\hat{s}_j = \sqrt{\mathrm{mean}(\mathbf{S}_j(\mathbf{y} - \sum_{k \neq j} \hat{\mathbf{f}}_k))}$ is the empirical estimate of $\sqrt{\mathbb{E}[(P_j R_j)^2]}$

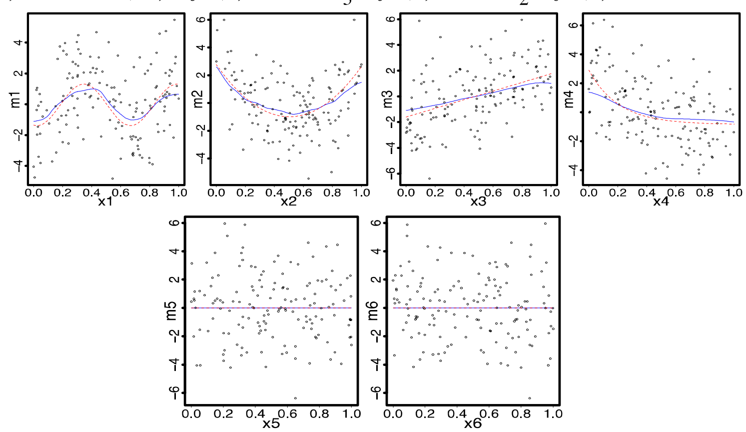
28

Example

- n =150, p = 200 (only 4 component functions are non-zeros)

$$Y_i = f_1(x_{i1}) + f_2(x_{i2}) + f_3(x_{i3}) + f_4(x_{i4}) + \epsilon_i$$

$$f_1(x) = -2\sin(2x), \;\; f_2(x) = x^2 - \frac{1}{3}, \;\; f_3(x) = x - \frac{1}{2}, \;\; f_4(x) = e^{-x} + e^{-1} - 1$$



Structured Sparse Additive Models (GroupSpAM)

GroupSpAM

- Exploit structured sparsity in the nonparametric setting
- The simplest structure is a non-overlapping group (or a partition of the original *p* variables)

$$\bigcup_{g \in \mathcal{G}} g = \{1, \ldots, p\} \text{ and } g \bigcap g' = \emptyset$$

- The optimization problem in the population setting is

$$\frac{1}{2}\mathbb{E}\left[\left(Y - \sum_{j=1}^{p} f_j(X_j)\right)^2\right] + \lambda \sum_{g \in \mathcal{G}} \sqrt{|g|} \sqrt{\sum_{j \in g} \mathbb{E}\left[f_j(X_j)^2\right]}$$

- Challenges:
  - New difficulty to characterize the thresholding condition at group level
  - No closed-form solution to the stationary condition, in the form of soft-thresholding step

31

Thresholding Conditions

- Theorem: the whole group *g* of functions $f_j = 0 \, \forall j \in g$ if and only if

$$\sqrt{\sum_{j \in g} \mathbb{E}[(P_j R_g)^2]} \leq \lambda \sqrt{|g|}$$

  - $R_g = Y - \sum_{g' \neq g} \sum_{j' \in g'} f_{j'}(X_{j'})$ is the partial residual after removing all functions from group *g*
  - Necessity: straightforward to prove
  - Sufficiency: more involved (see Yin et. al, 2012)

32

GroupSpAM Backfitting

Else,

Estimate $\hat{\mathbf{f}}_g$ by fixed point iteration,

$$\hat{\mathbf{f}}_g^{(t+1)} = \left(\widehat{\mathbf{J}} + \frac{\lambda\sqrt{|g|}}{\|\hat{\mathbf{f}}_g^{(t)}\|/\sqrt{n}} \mathbf{I} \right)^{-1} \widehat{\mathbf{Q}}\widehat{\mathbf{R}}_g.$$

Output: Fitted functions $\hat{\mathbf{f}} = \{\hat{\mathbf{f}}_j \in \mathbb{R}^n : j = 1, \ldots, p\}$.

Experiments

- Sample size n=150 and dimension p = 200, 1000
-

$f\ (\ )$		$2-\sin(x)$	
$f_4(x)$	$=$	$\exp(-x)$	8.98
$f_5(x)$	$=$	$x^3 + 1.5(x-1)^2$	14.57
$f_6(x)$	$=$	x	2.08
$f_7(x)$	$=$	$3\sin(\exp(-0.5x)$	0.80
$f_8(x)$	$=$	$-5\phi(x, 0.5, 0.8^2)$	3.76

Experiments (p = 200)

- Performance based on 100 independent simulations (t = 0)

method	precision	recall	$\#\hat{f}_1$	$\#\hat{f}_2$	$\#\hat{f}_3$	$\#\hat{f}_4$	$\#\hat{f}_5$	$\#\hat{f}_6$	$\#\hat{f}_7$	$\#\hat{f}_8$	MSE
GroupSpAM	1.00	1.00	100	100	100	100	100	100	100	100	7.22
SpAM	0.85	0.82	83	100	56	100	100	94	27	100	9.61
COSSO	0.66	0.42	6	1	27	100	50	61	3	88	28.29
GroupLasso	0.95	0.99	100	100	100	100	99	99	99	99	28.34

- Performance based on 100 independent simulations (t = 2)

method	precision	recall	$\#\hat{f}_1$	$\#\hat{f}_2$	$\#\hat{f}_3$	$\#\hat{f}_4$	$\#\hat{f}_5$	$\#\hat{f}_6$	$\#\hat{f}_7$	$\#\hat{f}_8$	MSE
GroupSpAM	0.89	0.99	100	100	100	100	98	98	98	98	7.26
SpAM	0.71	0.46	88	75	0	83	100	0	4	15	8.48
COSSO	0.23	0.41	11	61	22	90	76	10	10	47	13.72
GroupLasso	0.13	0.12	14	14	14	14	11	11	11	11	26.19

35

Experiments (p = 1000)

- Performance based on 100 independent simulations (t = 0)

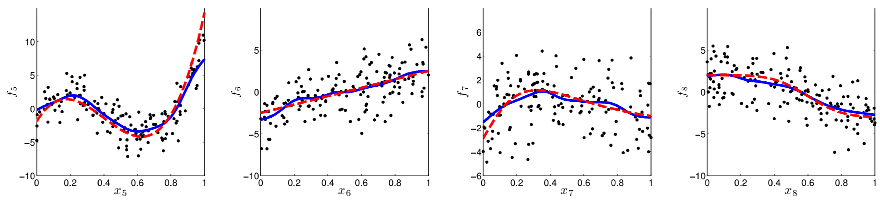
method	precision	recall	$\#\hat{f}_1$	$\#\hat{f}_2$	$\#\hat{f}_3$	$\#\hat{f}_4$	$\#\hat{f}_5$	$\#\hat{f}_6$	$\#\hat{f}_7$	$\#\hat{f}_8$	MSE
GroupSpAM	1.00	1.00	100	100	100	100	100	100	100	100	7.21
SpAM	0.86	0.68	49	91	25	100	100	71	7	97	11.66
COSSO	0.01	0.97	93	100	97	100	100	100	84	100	36.59
GroupLasso	0.93	0.97	98	98	98	98	97	97	97	97	29.49

- Performance based on 100 independent simulations (t = 2)

method	precision	recall	$\#\hat{f}_1$	$\#\hat{f}_2$	$\#\hat{f}_3$	$\#\hat{f}_4$	$\#\hat{f}_5$	$\#\hat{f}_6$	$\#\hat{f}_7$	$\#\hat{f}_8$	MSE
GroupSpAM	0.75	0.97	95	95	95	95	100	100	100	100	8.10
SpAM	0.69	0.34	59	43	0	65	100	0	1	3	9.69
COSSO	0.00	0.00	0	0	0	0	0	0	0	0	26.30
GroupLasso	0.02	0.03	4	4	4	4	2	2	2	2	25.86

36

Estimated Component Functions



GroupSpAM with Overlap

- Allow overlap between the different groups (Jacob et al., 2009)
- Idea: decompose each original component function to be a sum of a set of latent functions and then apply the functional group penalty to the decomposed

$$\text{subject to} \quad \sum_{g: j \in g} h_j^g = f_j, \; j = 1, \ldots, p.$$

  - The resulting support is a union of pre-defined groups
  - Can be reduced to the GroupSpAM with disjoint groups and solved by the same backfitting algorithm

Breast Cancer Data

- Sample size n = 295 tumors (metastatic vs non-metastatic) and dimension p = 3,510 genes.
- Goal: identify few genes that can predict the types of tumors.
- Group structure: each group consists of the set of genes in a pathway and groups are overlapping.

	GroupLasso	0.384	**44**	238
	GroupSpAM	0.358	**44**	**243**
2	SpAM	**0.349**	109	302
	GroupLasso	0.365	56	248
	GroupSpAM	**0.326**	**74**	149
3	SpAM	0.333	101	209
	GroupLasso	0.346	76	**138**

Summary

- Novel statistical method for structured functional sparsity in nonparametric additive models
  - Functional sparsity at the group level in additive models.
  - Can easily incorporate prior knowledge of the structures among the covariates.
  - Highly flexible: no assumptions are made on the design matrices or on the correlation of component functions in each group.
  - Benefit of group sparsity: better performance in terms of support recovery and prediction accuracy in additive models.

References

- Hastie, T. and Tibshirani, R. Generalized Additive Models. Chapman & Hall/CRC, 1990.
- Buja, A., Hastie, T., and Tibshirani, R. Linear Smoothers and Additive Models. Ann. Statist. Volume 17, Number 2 (1989), 453-510.
- Ravikumar, P., Lafferty, J., Liu, H., and Wasserman, L. Sparse additive models. JRSSB, 71(5):1009–1030, 2009.
- Yin, J., Chen, X., and Xing, E. Group Sparse Additive Models, ICML, 2012