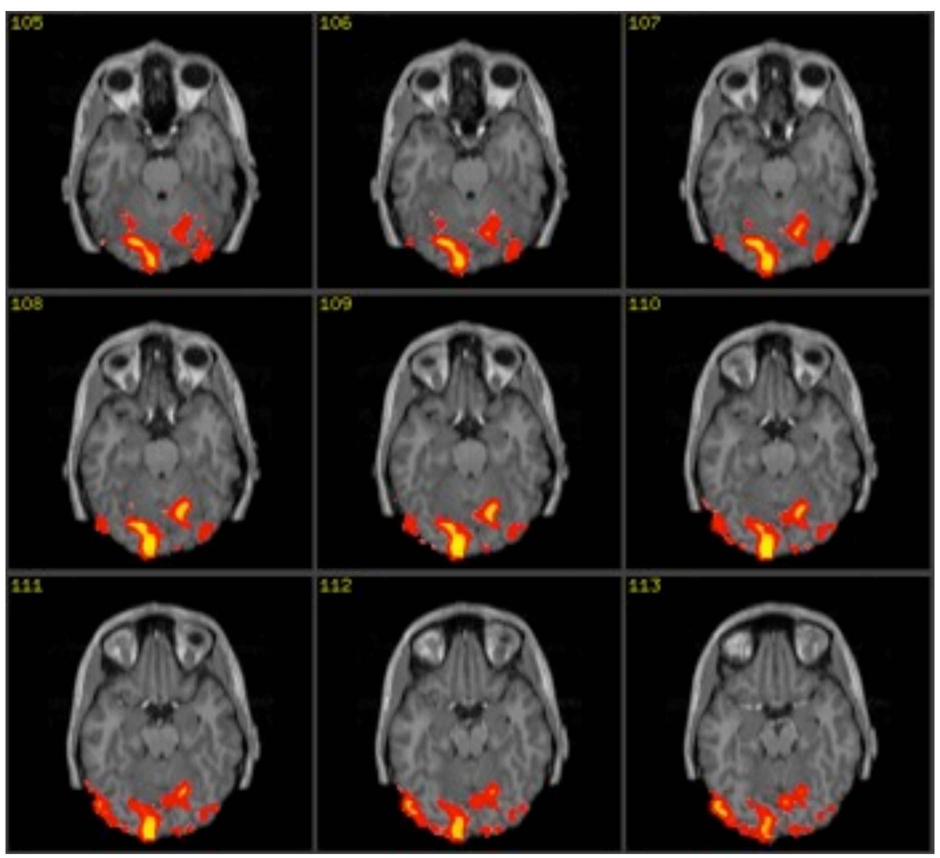
Single and Multiple Index Models

Pradeep Ravikumar
UT Austin

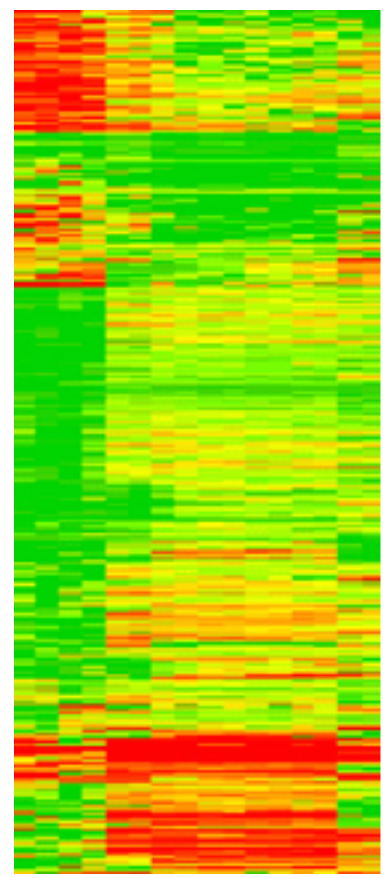
Joint work with X. Wang, M. Wainwright, B. Yu

Modern Data

- Across modern applications {images, signals, networks}
    - many^many variables in system than available observations



fMRI images



gene expression profiles

social networks

High-dimensional Data

- Curse of dimensionality
    - required observations/experience increase **exponentially** with variables in system
- Is there a way out?
    - Yes! If there is some intrinsic “structure” :: parameter lies in any of a collection of **low-dimensional subspaces** (Negahban, Ravikumar, Wainwright, Yu, 2009, 2012)

Examples of Structure Subspaces

Example 1. *Sparse vectors.* Consider the set of s-sparse vectors in p dimensions. For any particular subset $S \subseteq \{1, 2, \ldots, p\}$ with cardinality s, we define the model subspace

$$A(S) := \big\{\alpha \in \mathbb{R}^p \mid \alpha_j = 0 \quad \text{for all } j \notin S\big\}.$$

Example 2. *Group-structured norms.* In many applications, sparsity arises in a more structured fashion, with groups of coefficients likely to be zero (or non-zero) simultaneously. Suppose that $\{1, 2, \ldots, p\}$ can be partitioned into a set of T disjoint groups, say $\mathcal{G} = \{G_1, G_2, \ldots, G_T\}$. Given any subset $S_{\mathcal{G}} \subseteq \{1, \ldots, T\}$ of group indices, say with cardinality $s_{\mathcal{G}} = |S_{\mathcal{G}}|$, we can define the subspace

$$A(S_{\mathcal{G}}) := \big\{\alpha \in \mathbb{R}^p \mid \alpha_{G_t} = 0 \quad \text{for all } t \notin S_{\mathcal{G}}\big\}.$$

Example 3. *Low-rank matrices.* Consider the class of matrices $\Theta \in \mathbb{R}^{p_1 \times p_2}$ that have rank $r \leq \min\{p_1, p_2\}$. For any given matrix Θ, we let $\mathrm{row}(\Theta) \subseteq \mathbb{R}^{p_2}$ and $\mathrm{col}(\Theta) \subseteq \mathbb{R}^{p_1}$ denote its row space and column space respectively. For a given pair (U, V) of r-dimensional subspaces $U \subseteq \mathbb{R}^{p_1}$ and $V \subseteq \mathbb{R}^{p_2}$, we can define the subspaces $A(U, V)$ of $\mathbb{R}^{p_1 \times p_2}$ given by

$$A(U, V) := \big\{\Theta \in \mathbb{R}^{p_1 \times p_2} \mid \mathrm{row}(\Theta) \subseteq V,\ \mathrm{col}(\Theta) \subseteq U\big\}.$$

Negahban, Ravikumar, Wainwright, Yu, 2009, 2012

High-dimensional Data

- Curse of dimensionality
  - required observations/experience increase **exponentially** with variables in system
- Is there a way out?
  - Yes! If there is some intrinsic “structure” :: parameter lies in any of a collection of **low-dimensional subspaces** (Negahban, Ravikumar, Wainwright, Yu, 2009, 2012)
  - Such structure is typically focused on parametric models: e.g. **sparse** {Linear, Generalized Linear} Models, **low-rank** matrix-structured models, **edge-sparse** {Discrete, Gaussian} Graphical Models, ...

High-dimensional Data

- Curse of dimensionality
  - required observations/experience increase **exponentially** with variables in system
- Is there a way out?
  - Yes! If there is some intrinsic "structure" :: parameter lies in any of a collection of **low-dimensional subspaces** (Negahban, Ravikumar, Wainwright, Yu, 2009, 2012)
  - Such structure is typically focused on parametric models: e.g. **sparse** {Linear, Generalized Linear} Models, **low-rank** matrix-structured models, **edge-sparse** {Discrete, Gaussian} Graphical Models, ...
  - Non-parametric models: "Infinite" dimensional parameter-space, do not want to directly impose low-dimensional structure!

Semi-parametric Models

- Look at semi-parametric models with {parametric + non-parametric} components, and impose low-dimensional structure on the parametric component

Example: Additive Models

- General non-parametric regression model:

$$\underbrace{Y}_{\text{output}} = \underbrace{f(X_1, \ldots, X_p)}_{\text{signal}} + \text{noise}$$

- Additive Models: $Y = \sum_{j=1}^{p} f_j(X_j) + \epsilon$ (Hastie and Tibshirani, 90)
    - Sum of univariate functions of individual co-ordinates

Example: Additive Models

- General non-parametric regression model:

$$\underbrace{Y}_{\text{output}} = \underbrace{f(X_1, \ldots, X_p)}_{\text{signal}} + \text{noise}$$

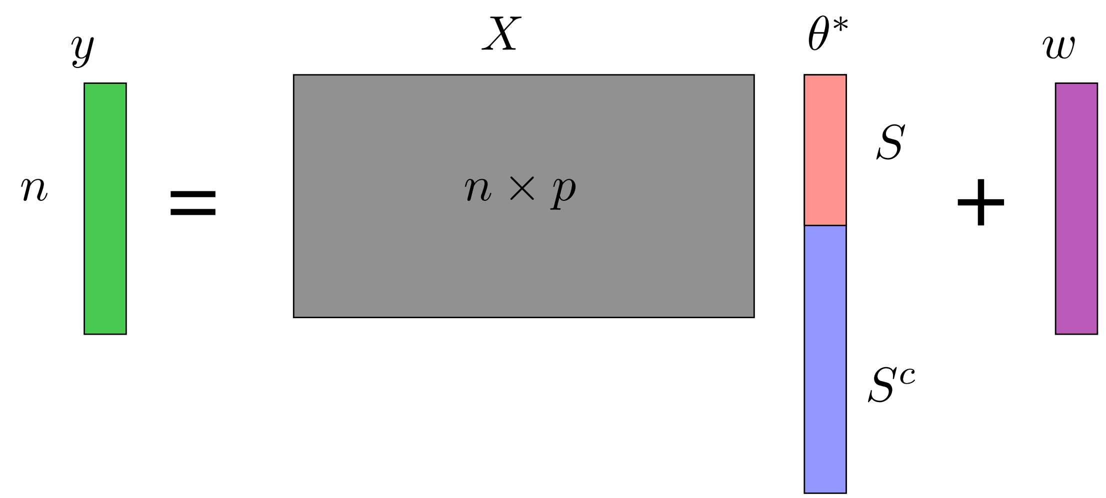
- Additive Models: $Y = \sum_{j=1}^{p} f_j(X_j) + \epsilon$ (Hastie and Tibshirani, 90)
  - Sum of univariate functions of individual co-ordinates
  - Rewrite as $Y = \sum_{j=1}^{p} \alpha_j \, g_j(X_j) + \epsilon$, with $\|g_j\| = 1, j = 1, \ldots, p$
  - Can impose low-dimensional structure on alpha

Example: Sparse Additive Models

- Additive Models: $Y = \sum_{j=1}^{p} f_j(X_j) + \epsilon$ (Hastie and Tibshirani, 90)
  - Rewrite as $Y = \sum_{j=1}^{p} \alpha_j \, g_j(X_j) + \epsilon, \text{ with } \|g_j\| = 1, j = 1, \ldots, p$
  - Impose sparsity on alpha ==> Sparse Additive Models (Ravikumar, Lafferty, Liu, Wasserman 07, Lin and Zhang 06, Meir, Van de Geer, Buhlmann 09, Raskutti, Wainwright, Yu 10, ...)
  - Other structured-sparse extensions (Liu et al. 2010, ...)
    - Group-sparse additive models, structured-sparse additive models, ...

Semi-parametric story only goes so far

Sparse Models



Set-up: noisy observations $y = X\theta^* + w$ with sparse θ^*

Estimator: Lasso program

$$\widehat{\theta} \in \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^T \theta)^2 + \lambda_n \sum_{j=1}^{p} |\theta_j|$$

Some past work: Tibshirani, 1996; Chen et al., 1998; Donoho/Xuo, 2001; Tropp, 2004; Fuchs, 2004; Meinshausen/Buhlmann, 2005; Candes/Tao, 2005; Donoho, 2005; Haupt & Nowak, 2006; Zhao/Yu, 2006; Wainwright, 2006; Zou, 2006; Koltchinskii, 2007; Meinshausen/Yu, 2007; Tsybakov et al., 2008

Sparse Nonparametric Models

$$Y = \sum_{j=1}^{p} f_j(X_j) + \epsilon,$$

$$|\{j \in [p] : f_j \neq 0\}| \ll p$$

Sparse Additive Models can be rewritten as a semi-parametric model as noted before

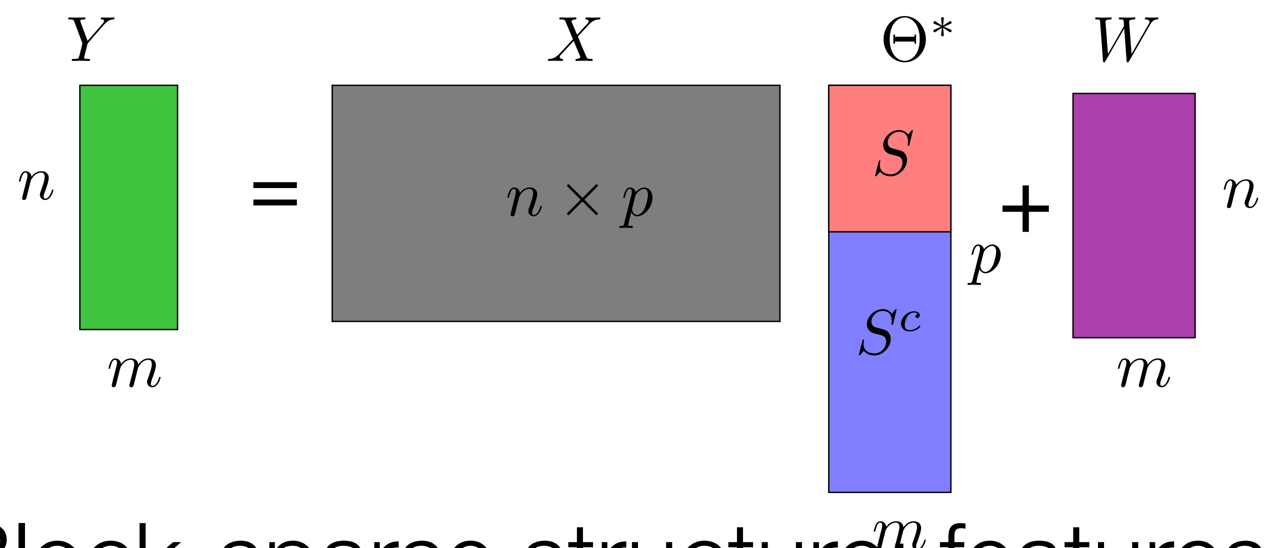
Sparse Nonparametric Models

$$Y = f(X_1, \ldots, X_p) + \epsilon,$$
$$|\{j \in [p] : f(\cdot) \text{ depends on } X_j| \ll p$$

Liu, Lafferty, Wasserman 06; Bertin, Lecue 08

Not easily rewritten as a semi-parametric model

Block-sparse Models



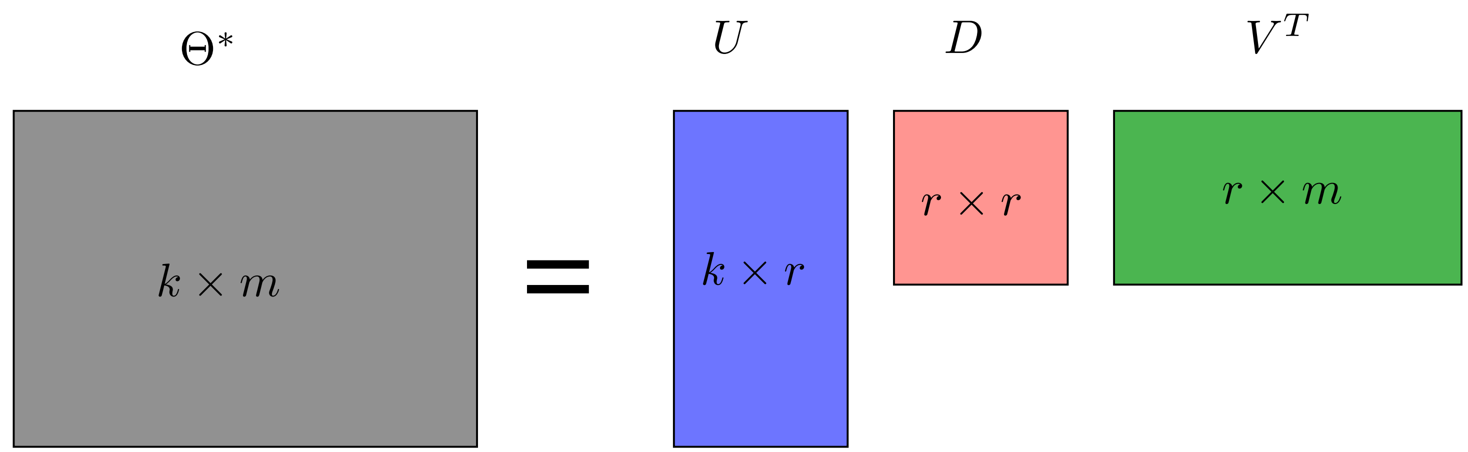
Block-sparse structure: features (rows) shared across tasks (columns)

Group LASSO (Obozinski et al; Negahban et al; Huang et al)

$$\min_{\beta} \sum_{k=1}^{r} \frac{1}{n_k} \sum_{i=1}^{n_k} \left\| y_i^{(k)} - X_i^{(k)} \beta^{(k)} \right\|_2^2 + \lambda \left\| \beta \right\|_{1,\infty}$$

$$\|\beta\|_{1,\infty} = \sum_j \max_k \left| \beta_j^{(k)} \right|$$

Low-rank Models



Set-up: Matrix $\Theta^* \in \mathbb{R}^{k \times m}$ with rank $r \ll \min\{k, m\}$.

Estimator:

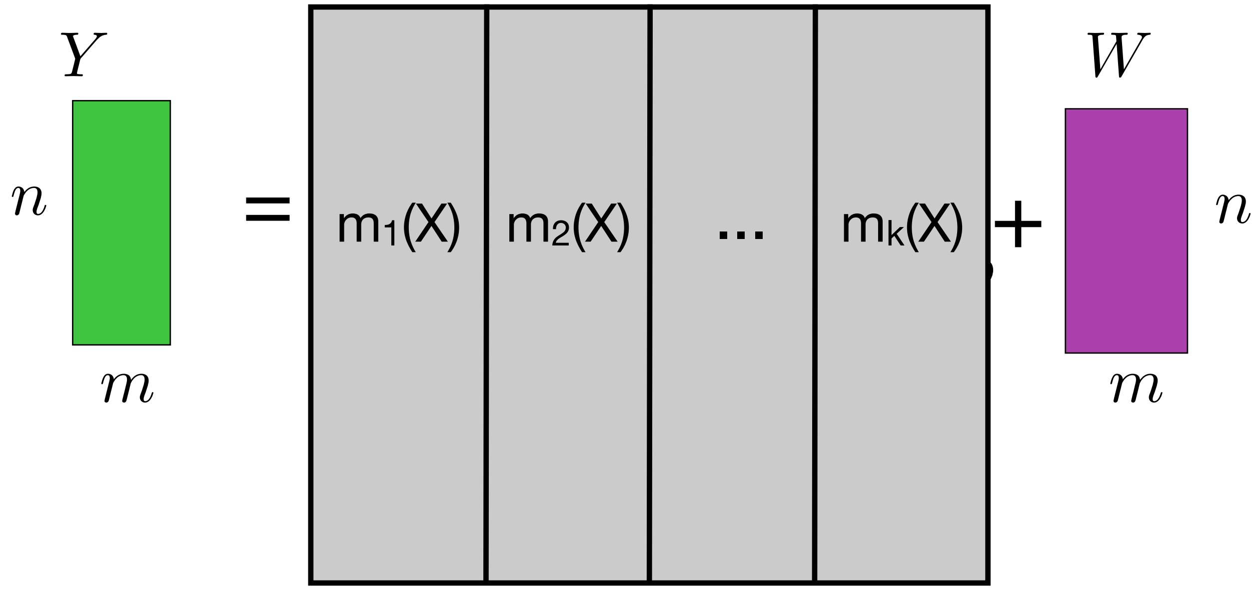
$$\widehat{\Theta} \in \arg\min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle\langle X_i,\ \Theta \rangle\rangle)^2 + \lambda_n \sum_{j=1}^{\min\{k,m\}} \sigma_j(\Theta)$$

Some past work: Frieze et al., 1998; Achilioptas & McSherry, 2001; Srebro et al., 2004; Drineas et al., 2005; Rudelson & Vershynin, 2006; Recht et al., 2007; Bach, 2008; Meka et al., 2009; Candes & Tao, 2009; Keshavan et al., 2009

Nonparametric Low-Rank Models

- Not even obvious what the corresponding structure in the non/semi-parametric case would be

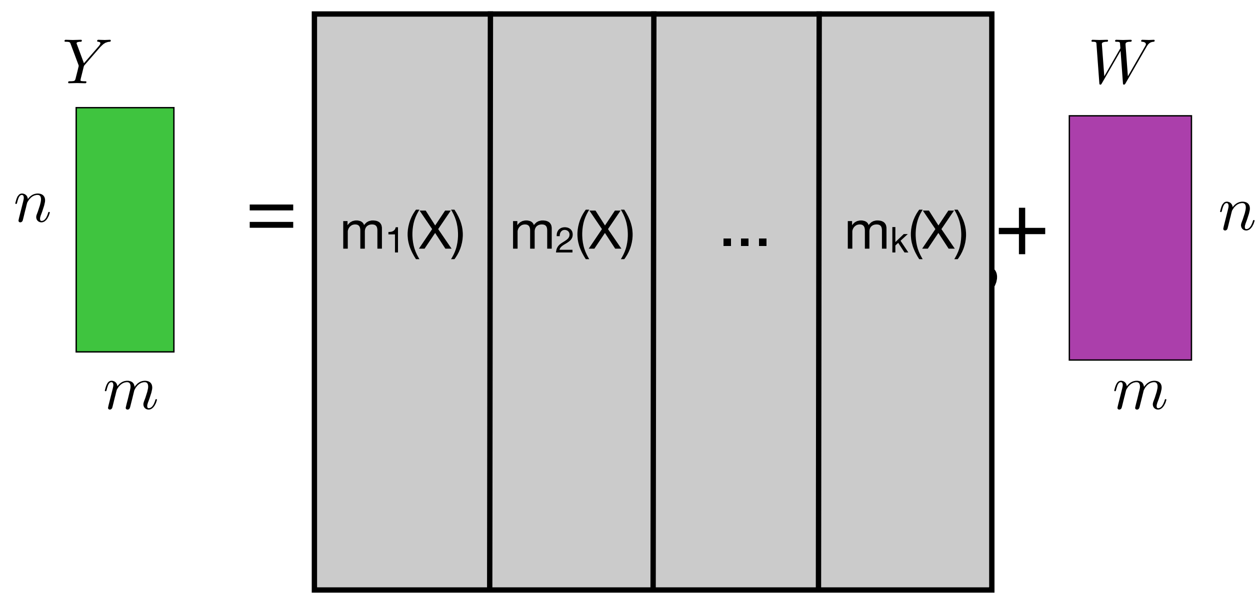
- Foygel et al. 2012:



Nonparametric Low-Rank Models

- Not even obvious what the corresponding structure in the non/semi-parametric case would be

- Foygel et al. 2012:



$$\mathrm{Cov}(m(X)) \text{ has low rank}$$

> A unified story for non-parametric structure (akin to Negahban et al., 2009, 2012 for parametric structure) is still outstanding

> More than imposing parametric structure on a semi-parametric model

Multiple Index Model

Response **Y** as a function of the dependent variables **X**:

$$Y = \sum_{j=1}^{m} g_j(\beta_j^T X) + \epsilon,$$

Multiple Index Model

Response **Y** as a function of the dependent variables **X**:

$$Y = \sum_{j=1}^{m} g_j(\boxed{\beta_j^T X}) + \epsilon,$$

“Index” :: a uni-dimensional summary of data

Multiple Index Model

Response **Y** as a function of the dependent variables **X**:

$$Y = \sum_{j=1}^{m} \boxed{g_j(\beta_j^T X)} + \epsilon,$$

"component"

Also called a ridge function

- $g_j(\beta_j^T X)$ is constant where $\beta_j^T X$ is constant
- Its function surface looks like a ridge

Multiple Index Model

Response **Y** as a function of the dependent variables **X**:

$$Y = \sum_{j=1}^{m} g_j(\beta_j^T X) + \epsilon,$$

- Task: Given **n** samples (X^i, Y^i), recover the functions $\{g_j\}_{j=1}^m$ and the weights $\{\beta_j\}_{j=1}^m$
  - Can impose {sparsity, other low-dimensional structure} on scales of g_j (like in sparse additive models), as also on \beta_j

Multiple Index Model

Response **Y** as a function of the dependent variables **X**:

$$Y = \sum_{j=1}^{m} g_j(\beta_j^T X) + \epsilon,$$

- Task: Given **n** samples (X^i, Y^i), recover the functions $\{g_j\}_{j=1}^m$ and the weights $\{\beta_j\}_{j=1}^m$
  - Can impose {sparsity, other low-dimensional structure} on scales of g_j (like in sparse additive models), as also on \beta_j
  - For now, consider vanilla multiple index models

Occurrences in the wild

Response **Y** as a function of the dependent variables **X**:

$$Y = \sum_{j=1}^{m} g_j(\beta_j^T X) + \epsilon,$$

- Neural networks: functions $\mathbf{g_j}$ set to sigmoids
- Modeling Distributions over images: product (instead of sum) of such functions (Hinton, 99; Roth, Black, 05; Welling, Hinton, Osindero, 02)

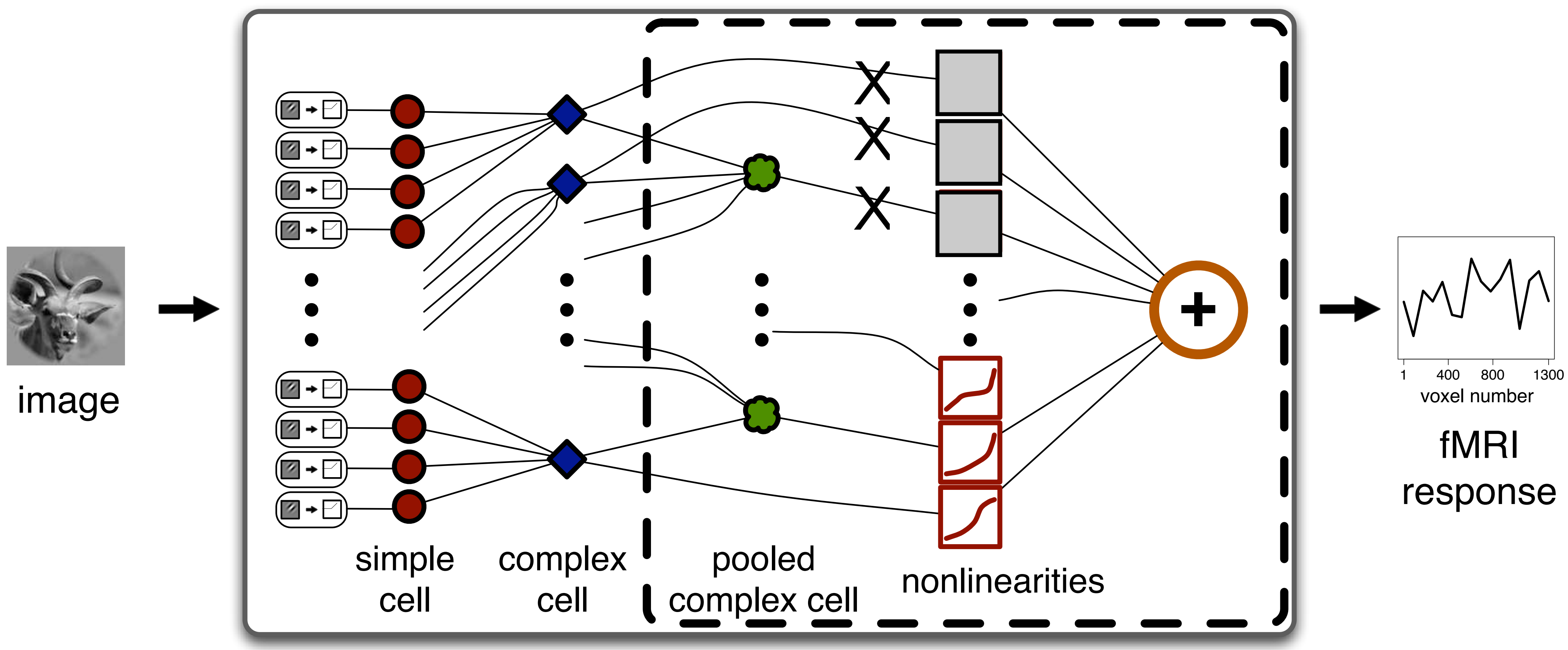
Application: Neural Coding

- Neural Coding: how neurons process and encode information
- Typical models use linear filters on the visual stimulus
  - easy to fit to data, computationally tractable, fits observed responses of neurons in “early” sensory areas
- But non-linear sub-units play a key role
  - Experiments demonstrating presence of non-linear units in visual cortex date to ’76 and earlier (Hochstein, Shapely 76)
  - Even canonical “simple” cells have non-linearities (Rust et al. 05, Touryan et al. 05)

Application: Responses in early visual cortex (V1)

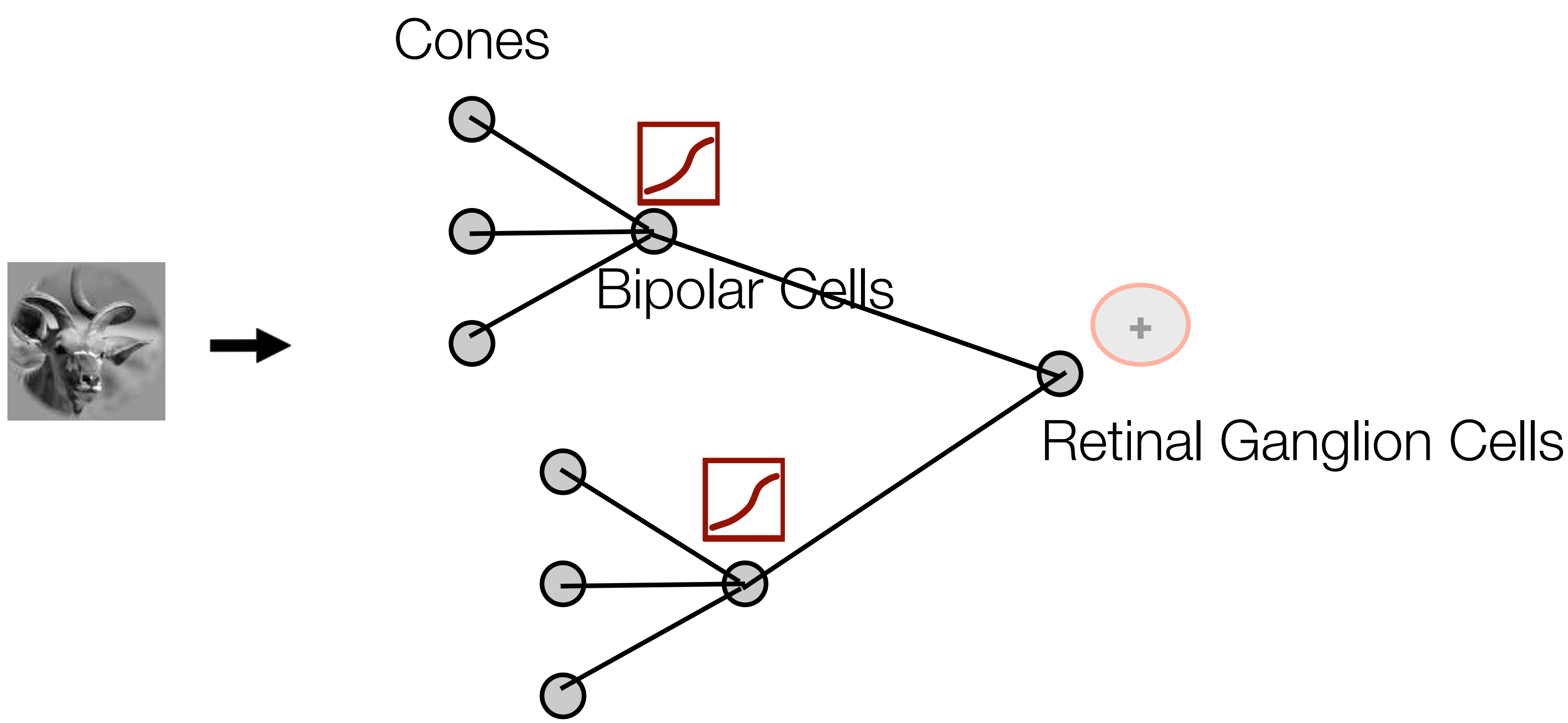
Used sparse additive models to encode voxels in early visual cortex

Encoding: Ravikumar et al. 2009
Decoding: Vu et al. 2010



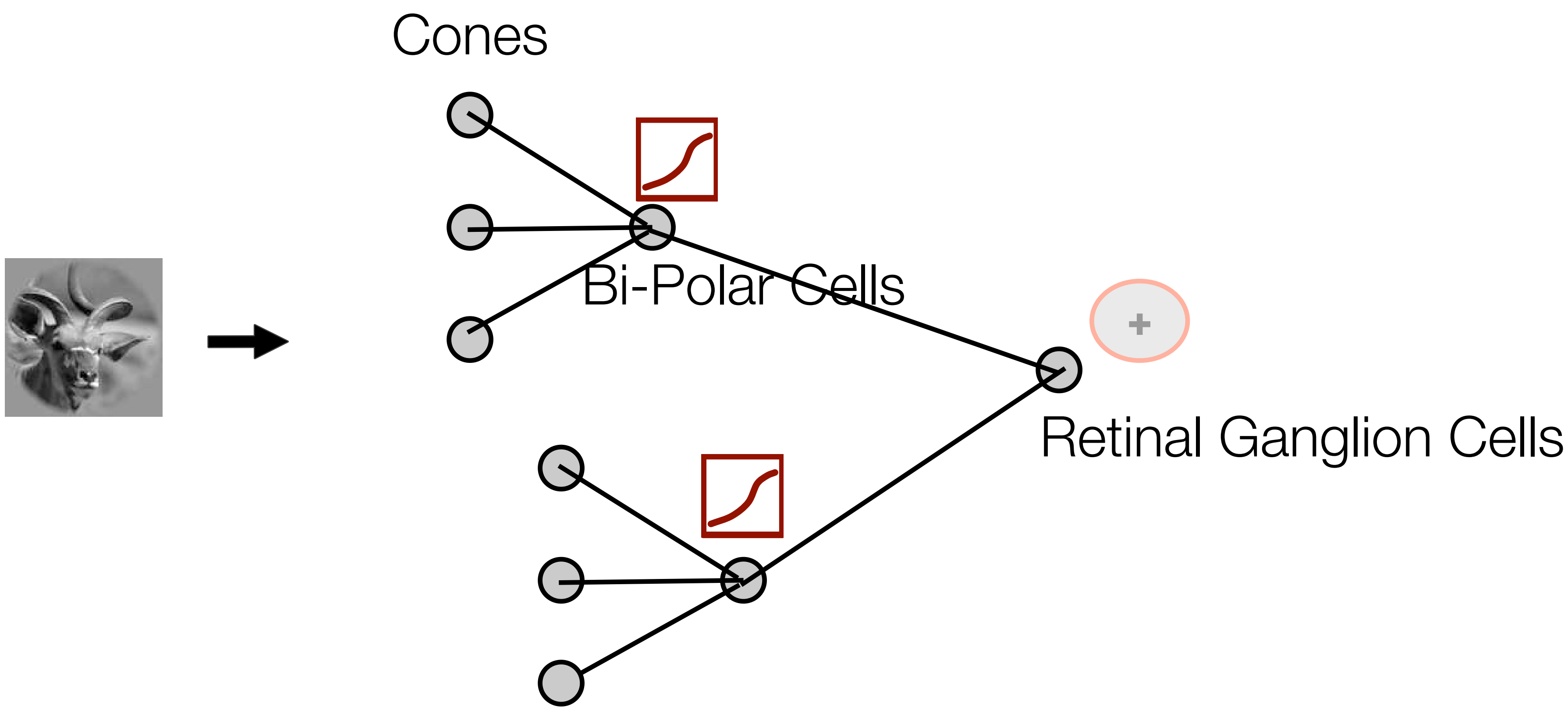
Application: Retinal Modeling

- Cone cells, feed into bipolar cells, which feed into Retinal Ganglion Cells



- It is possible to record Retinal Ganglion Cells in response to visual stimuli, but difficult to record from, and consequently infer the statistical behavior of bi-polar cells

Multiple Index Models



$$Y = \sum_{j=1}^{m} g_j(\beta_j^T X) + \epsilon,$$

To the Rescue!

Index Models and Projections

- When data is high-dimensional, then for {visualization, modeling}, a classical technique is based on
  - (a) projecting data into lower dimensional space, and
    (b) working with projected data
- Salient Question: How to pick the projection directions?
  - Friedman: Visualization; inspect 2D projections
  - Huber: Interestingness
    - PCA, ICA, methods by Kruskal, Switzer and Wright, ...
    - Friedman, Tukey 74: max. product of density and std-dev of projected data

On Index Models and Projections

- Multiple Index Models: Additive Models on Projected Data

- Additive Models: $Y = \sum_{j=1}^{p} f_j(X_j) + \epsilon$ (Hastie and Tibshirani, 90)
    - Sum of univariate functions of individual co-ordinates

- Multiple Index Models:
    - Indices formed by projections $\{Z_j = \beta_j^T X\}$
    - Additive Model over indices:

$$\begin{aligned} Y &= \sum_j g_j(Z_j) \\ &= \sum_j g_j(\beta_j^T X) \end{aligned}$$

Projection Pursuit Regression

- Candidate Criterion for picking "interesting" projection directions in multiple index model
  - Minimize squared error
- Projection Pursuit Regression (Friedman and Stuetzle, 81)
  - Minimize squared error greedily

Backfitting

- Additive Models typically inferred using “backfitting”
  - Cycle through coordinates, and fit univariate function in that co-ordinate to the residual
  - Can extend back-fitting to multiple-index models

Multiple Index Model Backfitting

$$\min_{\{\beta_j \in \mathbb{R}^{|I_j|}, g_j \in \mathcal{G}\}} \frac{1}{2n} \sum_{i=1}^{n} (Y^{(i)} - \sum_{j=1}^{m} g_j(\beta_j^T X_j^{(i)}))^2$$

Algorithm Least-Squares Multiple-Index Backfitting

Initialize: $\beta_j = 0$, $g_j = 0$; $j = 1, \ldots, m$.
for outer iterations $t = 1, 2, \ldots$ until convergence **do**
  for $k = 1, \ldots, m$ **do**
    Compute the residuals $R_k^{(i)} = Y^{(i)} - \sum_{j \neq k} g_j(\beta_j^T X_j^{(i)})$; $i = 1, \ldots, n$.
    Solve for (g_k, β_k) by estimating a sparse single-index model with R_k as output and X_k as input.
  end for
end for

Multiple Index Model Backfitting

$$\min_{\{\beta_j \in \mathbb{R}^{|I_j|}, g_j \in \mathcal{G}\}} \frac{1}{2n} \sum_{i=1}^{n} (Y^{(i)} - \sum_{j=1}^{m} g_j(\beta_j^T X_j^{(i)}))^2$$

Algorithm Least-Squares Multiple-Index Backfitting

Initialize: $\beta_j = 0$, $g_j = 0$; $j = 1, \dots, m$.
for outer iterations $t = 1, 2, \dots$ until convergence **do**
  for $k = 1, \dots, m$ **do**
    Compute the residuals $R_k^{(i)} = Y^{(i)} - \sum_{j \neq k} g_j(\beta_j^T X_j^{(i)})$; $i = 1, \dots, n$.
    Solve for (g_k, β_k) by estimating a sparse single-index model with R_k as output and X_k as input.
  end for
end for

Estimating a SIM model is key!

Candidate Method for SIM Estimation

$$Y^{(i)} = g(\beta^T X^{(i)}) + \epsilon$$

Algorithm Solving a single-index model

Initialize: $\beta = 0$, $g = 0$.

for outer iterations $t = 1, 2, \ldots$ until convergence **do**

Fixing g, obtain β by solving:

$$\beta \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (Y^{(i)} - g(\beta^T X^{(i)}))^2 \right\}.$$

Fixing β, obtain g by solving

$$g \in \arg\min_{g \in \mathcal{G}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (Y^{(i)} - g(\beta^T X^{(i)}))^2 \right\}.$$

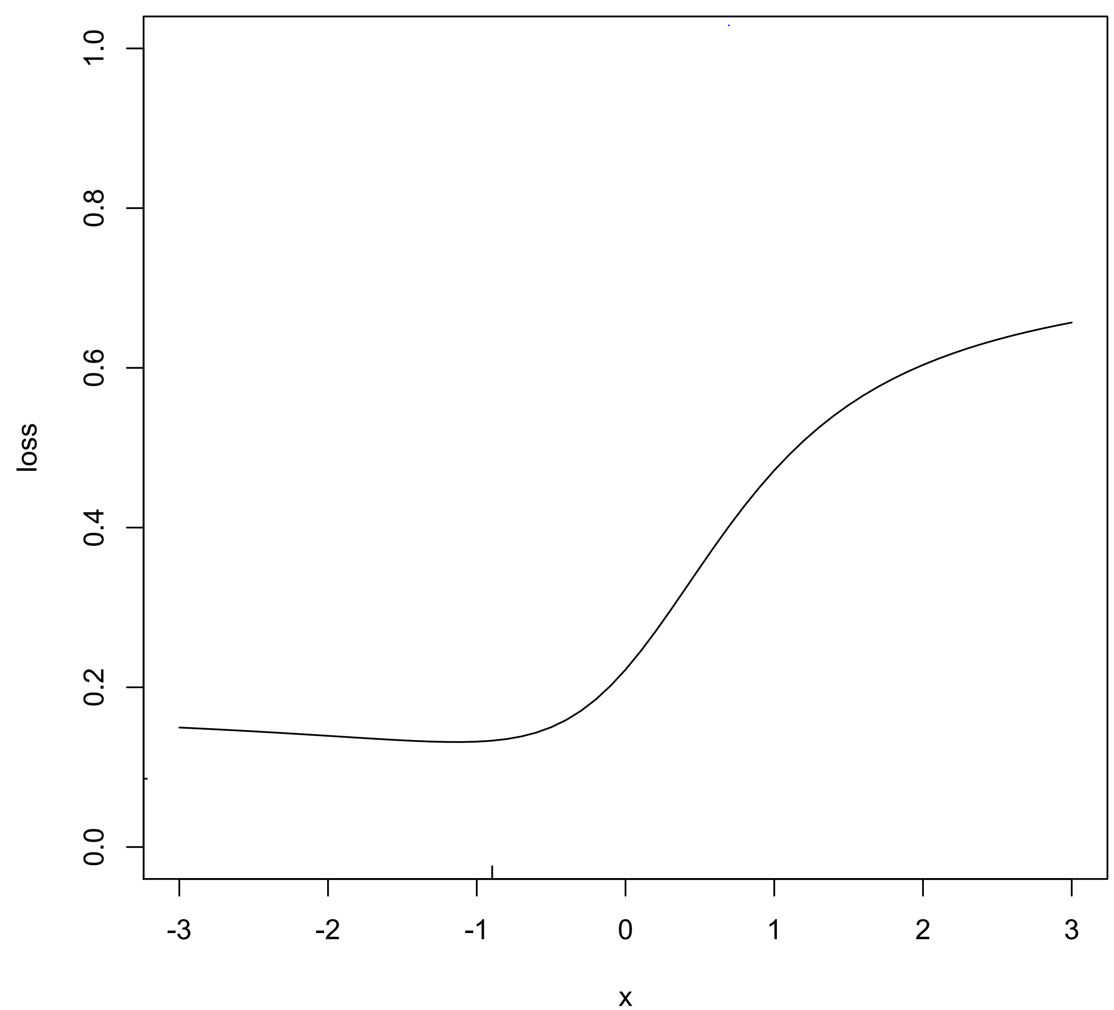
end for

Step II in SIM estimation: Fitting the Proj. Weights

- Consider loss, as a function of beta, fixing g

$$L(\beta) = \mathbb{E}(Y - g(\beta^T X))^2$$

- 1D Example

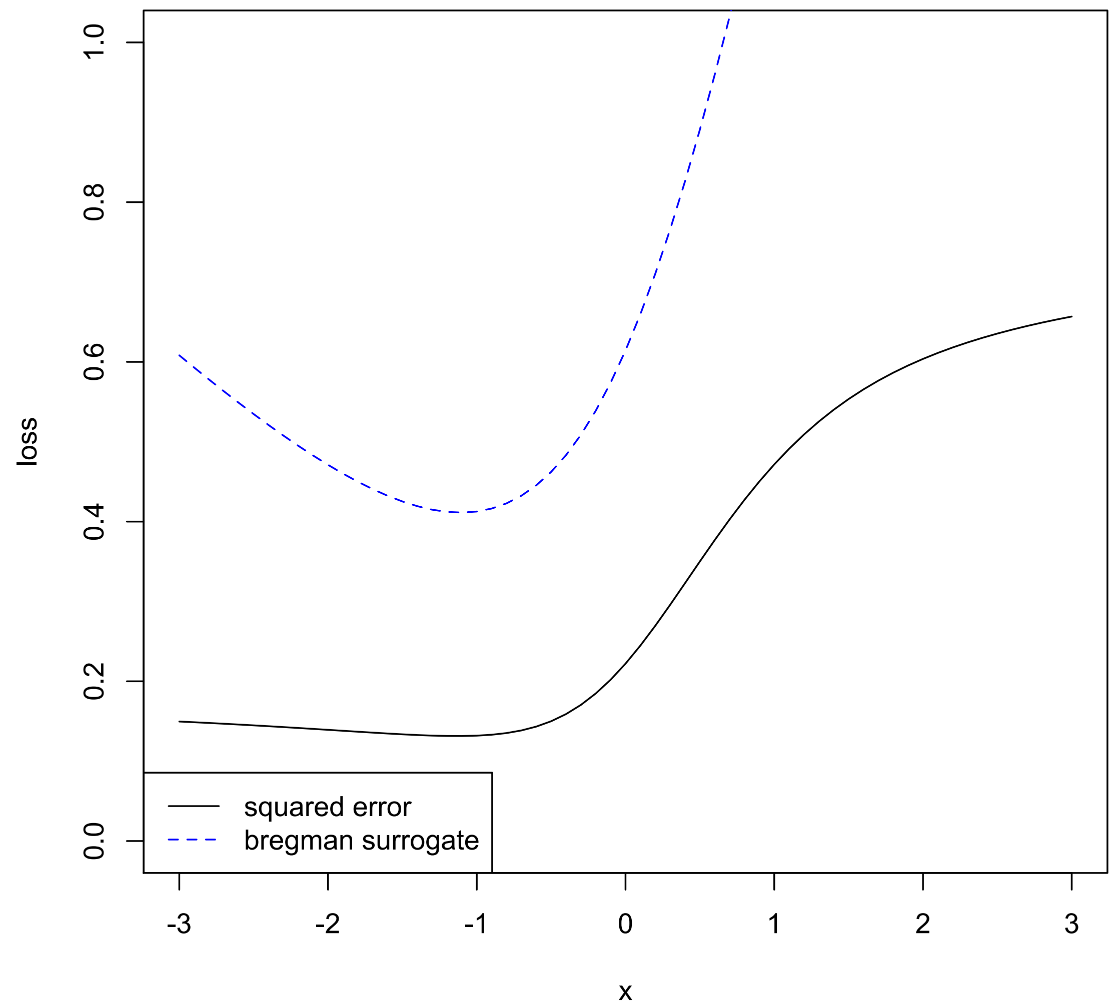


Single Index Model Loss

- Consider loss, as a function of beta, fixing g

$$L(\beta) = \mathbb{E}(Y - g(\beta^T X))^2$$

- 1D Example

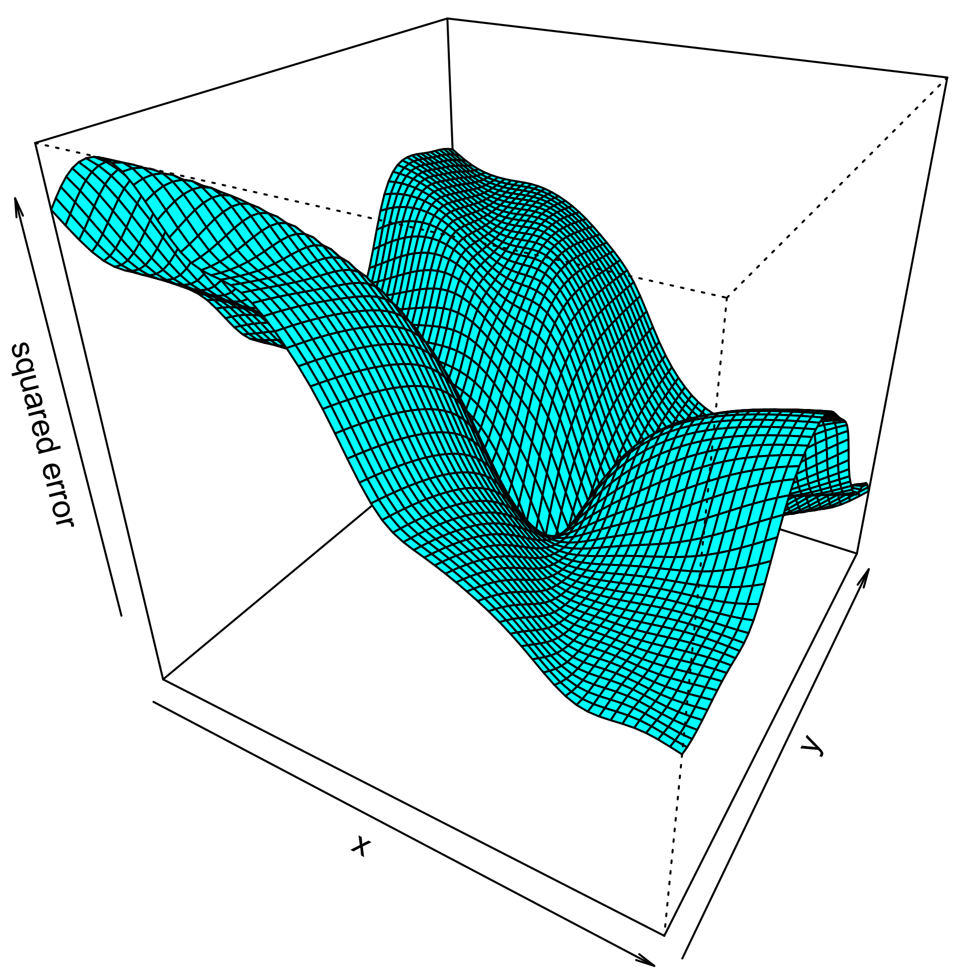


Single Index Model Loss

- Consider loss, as a function of beta, fixing g

$$L(\beta) = \mathbb{E}(Y - g(\beta^T X))^2$$

- 2D Example

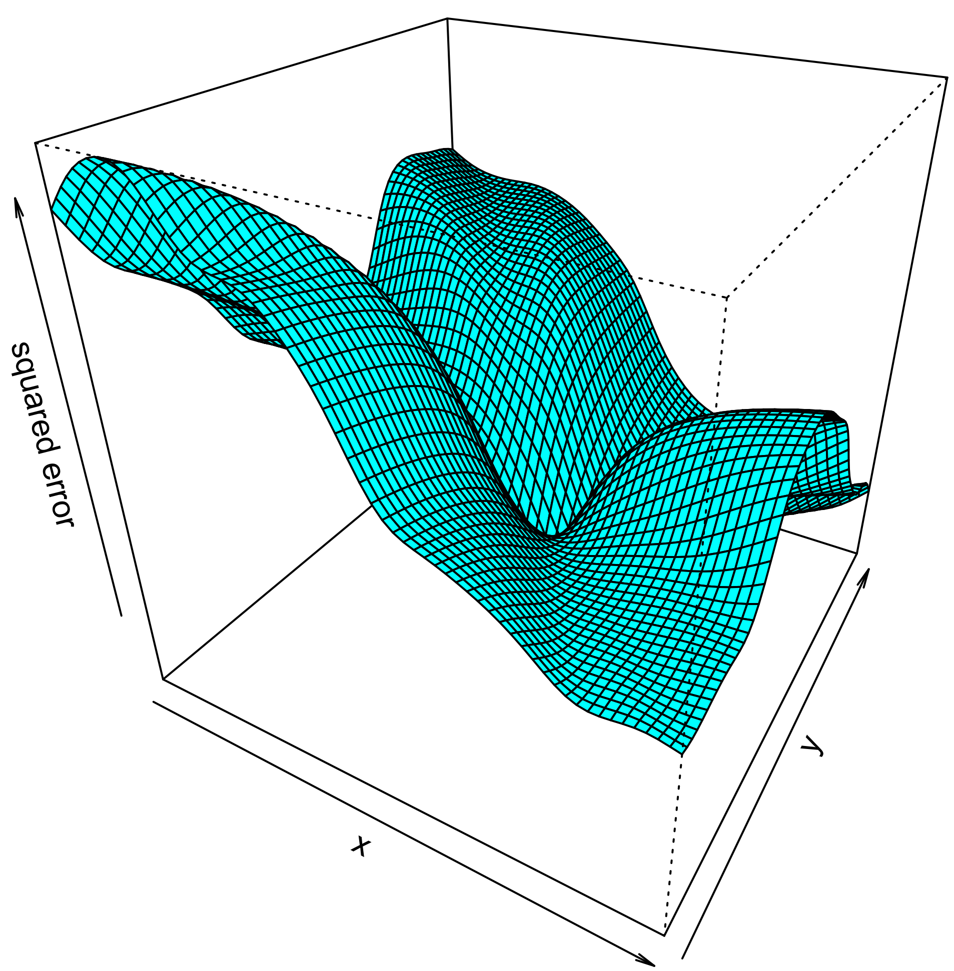


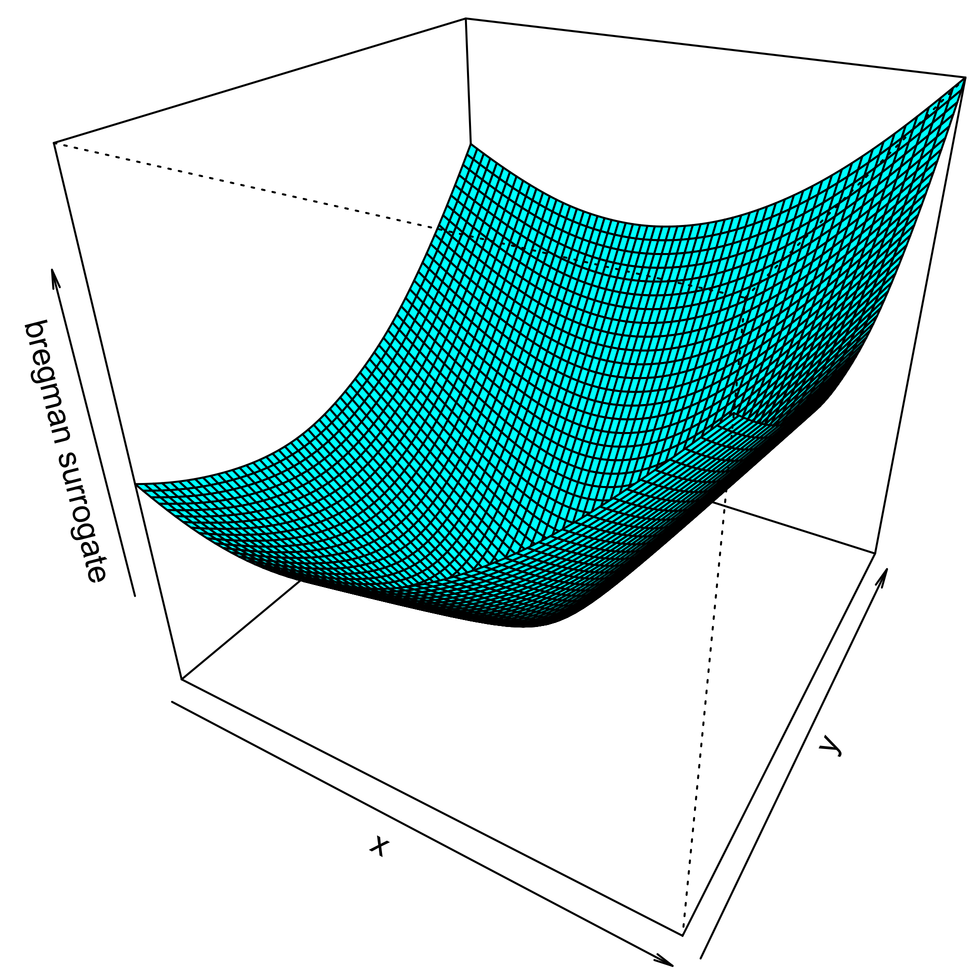
Single Index Model Loss

- Consider loss, as a function of beta, fixing g

$$L(\beta) = \mathbb{E}(Y - g(\beta^T X))^2$$

- 2D Example





A surrogate loss

- The squared error loss $\mathbb{E}(Y - g(\beta^T X))^2$ is a notion of divergence between $Y \text{ and } g(\beta^T X)$

- Are there are other loss functions, that

  - (a) arise as measuring divergence between $Y \text{ and } g(\beta^T X)$, but are also

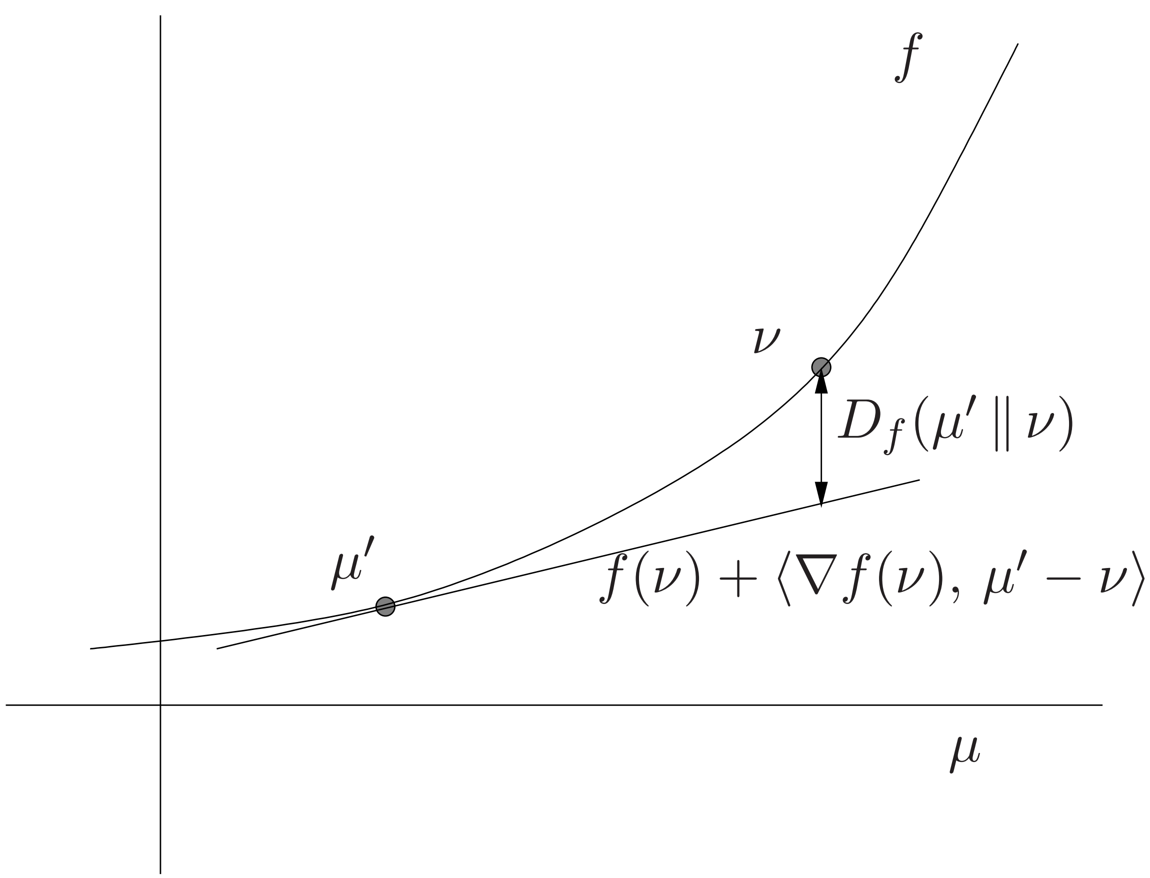
  - (b) convex in beta

- Yes!

Bregman Divergence

- Given a strictly convex function f, the induced Bregman divergence:

$$D_f(\mu' \,\|\, \nu) \quad := \quad f(\mu') - f(\nu) - \langle \nabla f(\nu),\, \mu' - \nu \rangle$$



- Euclidean Distance ::

  With $f(u) = u^2$, $D_f(\mu' \| \nu) = \|\mu' - \nu\|_2^2$

Surrogate Bregman Loss

- Squared Error Loss, as a function of beta, fixing g

$$L(\beta) = \mathbb{E}(Y - g(\beta^T X))^2$$

- Let $G(v) = \int_{\infty}^{v} g(t)dt,$ and $F(u) = \sup_{v \in \mathbb{R}} v^T u - G(v),$

- **Proposition**:

$$D_F(Y \,\|\, g(\beta^T X) = G(\beta^T X) - \beta^T X\, Y + F(Y)$$

is convex in beta, when g is monotonic.

SIM Estimation using Surrogate Bregman Loss

Algorithm Solving a single-index model: Bregman Updates

Initialize: $\beta = 0$, $g = 0$.

for outer iterations $t = 1, 2, \ldots$ until convergence **do**

Fixing g, obtain β by solving:

$$\beta \in \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \Big(G(\beta^T X^{(i)}) - Y^{(i)}(\beta^T X^{(i)}) \Big) \right\}.$$

Fixing β, obtain g by solving

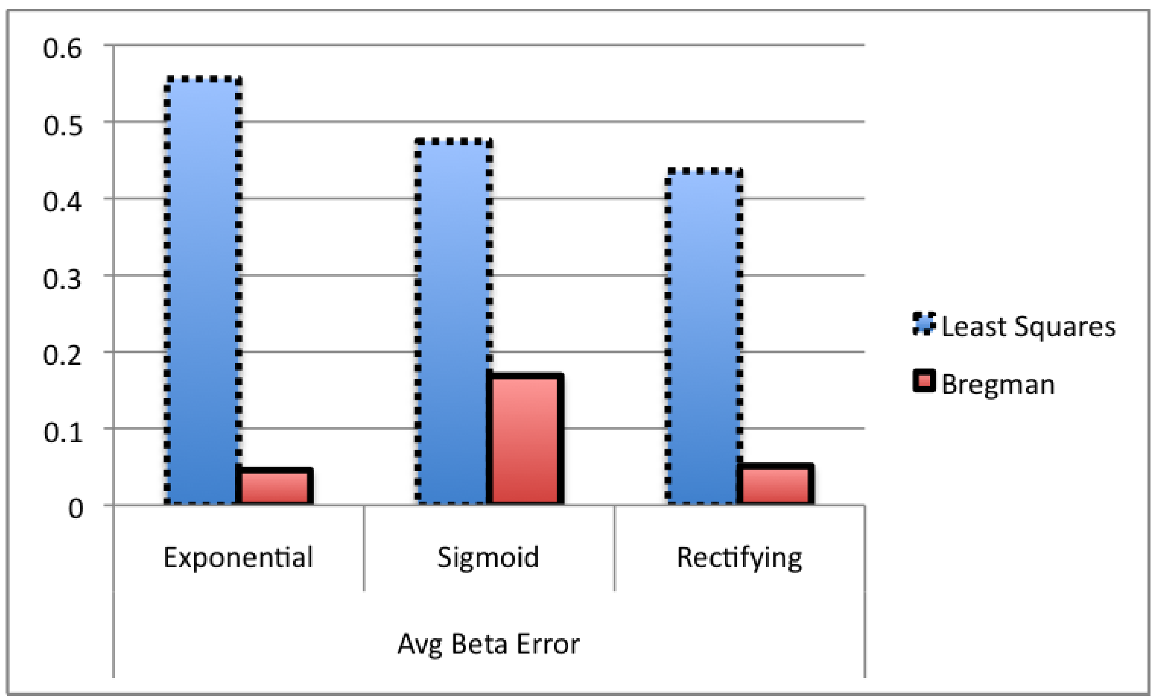
$$g \in \arg\min_{g \in \mathcal{G}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (Y^{(i)} - g(\beta^T X^{(i)}))^2 \right\}.$$

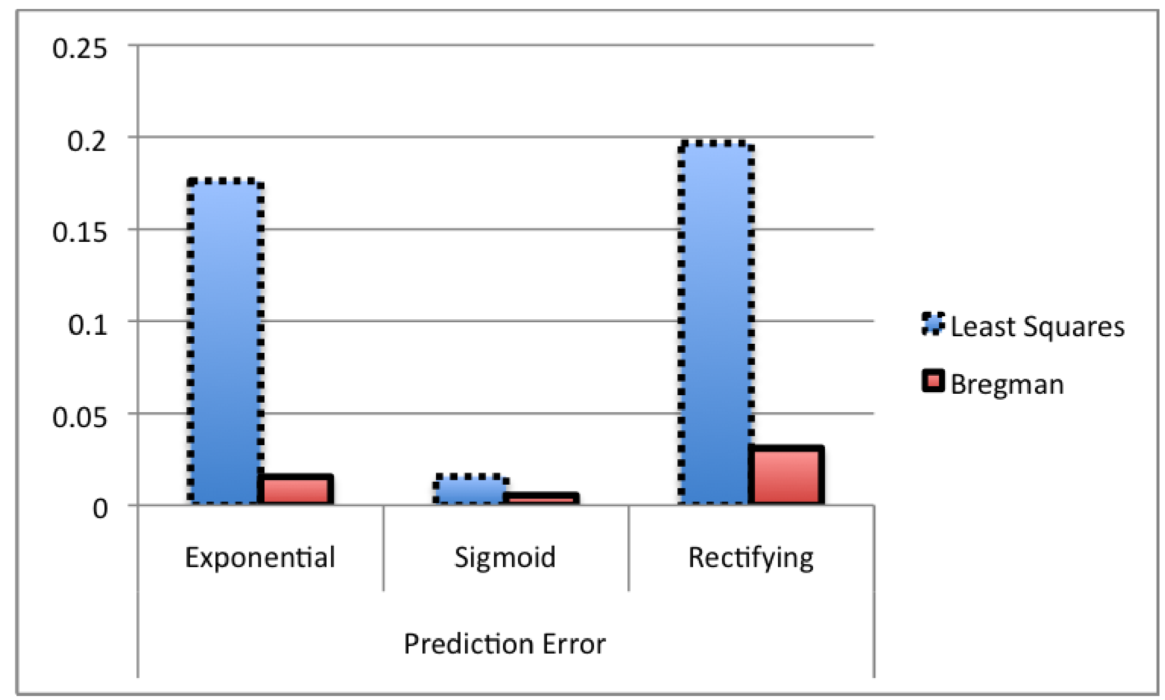
end for

Application: Retinal Modeling

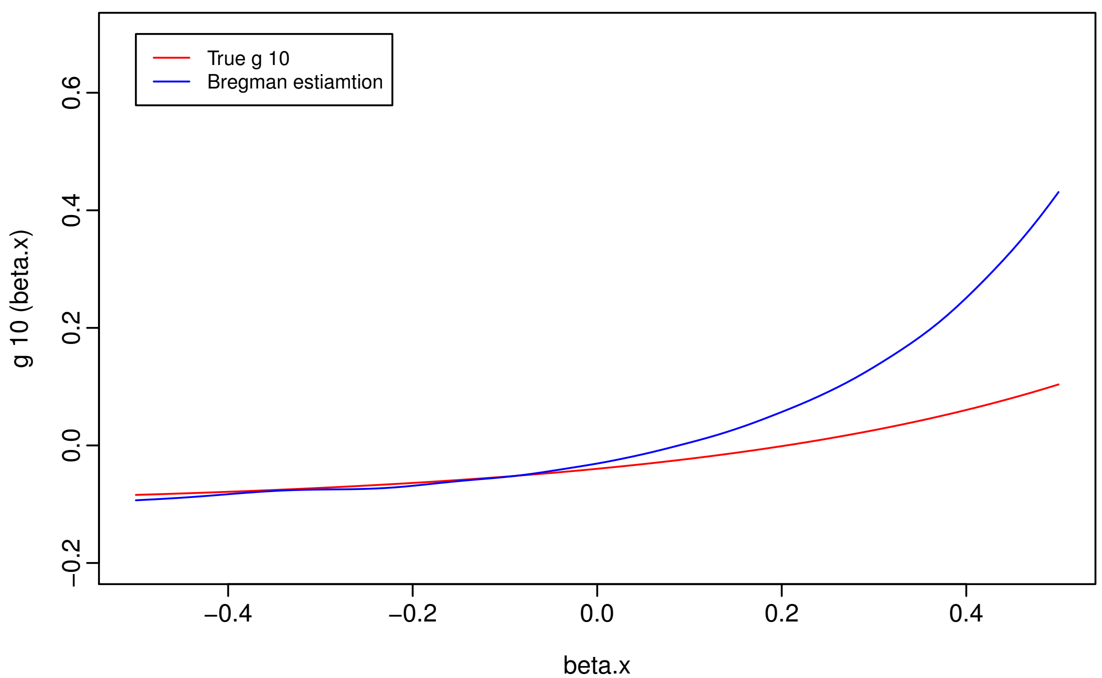
- Simulations of {cones, bi-polar cells, retinal ganglion cells} from Chichilnisky Lab
  - Corresponding to 48015 visual (white noise) stimuli:
  - Simulated responses of 134 cones, subsets of which provide input to 20 bipolar cells that feed into a single retinal ganglion cell.
  - Code allows us to fix the nonlinearity in bipolar cell outputs
    - We use exponential, sigmoidal, rectifying (hinge) functions

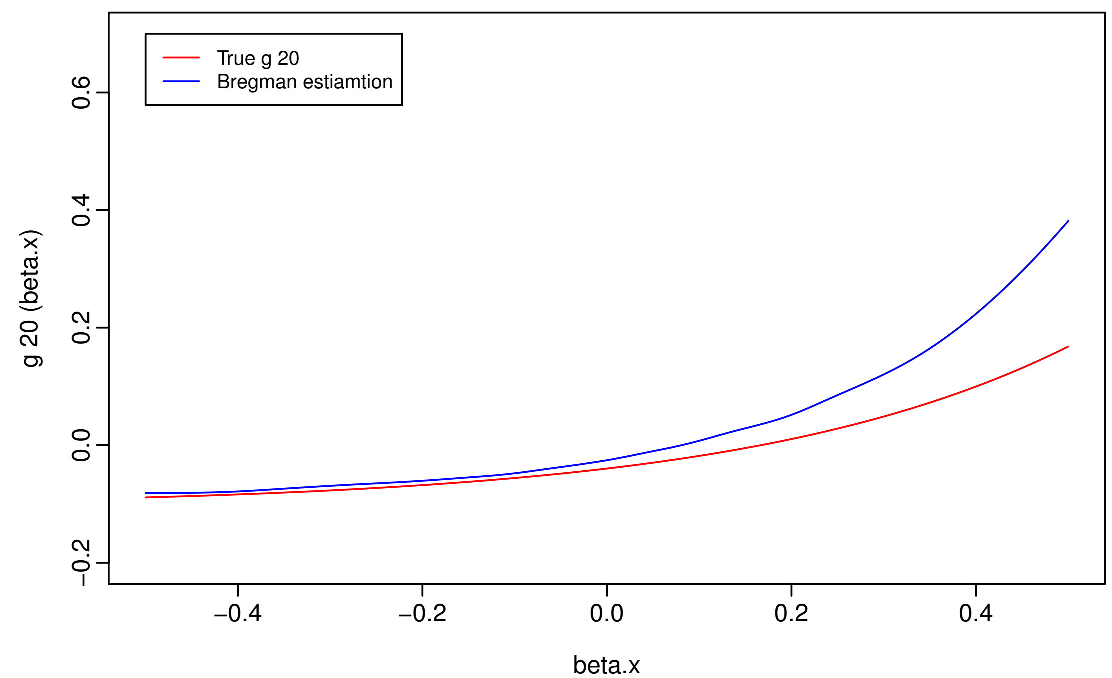
Parameter and Prediction Error



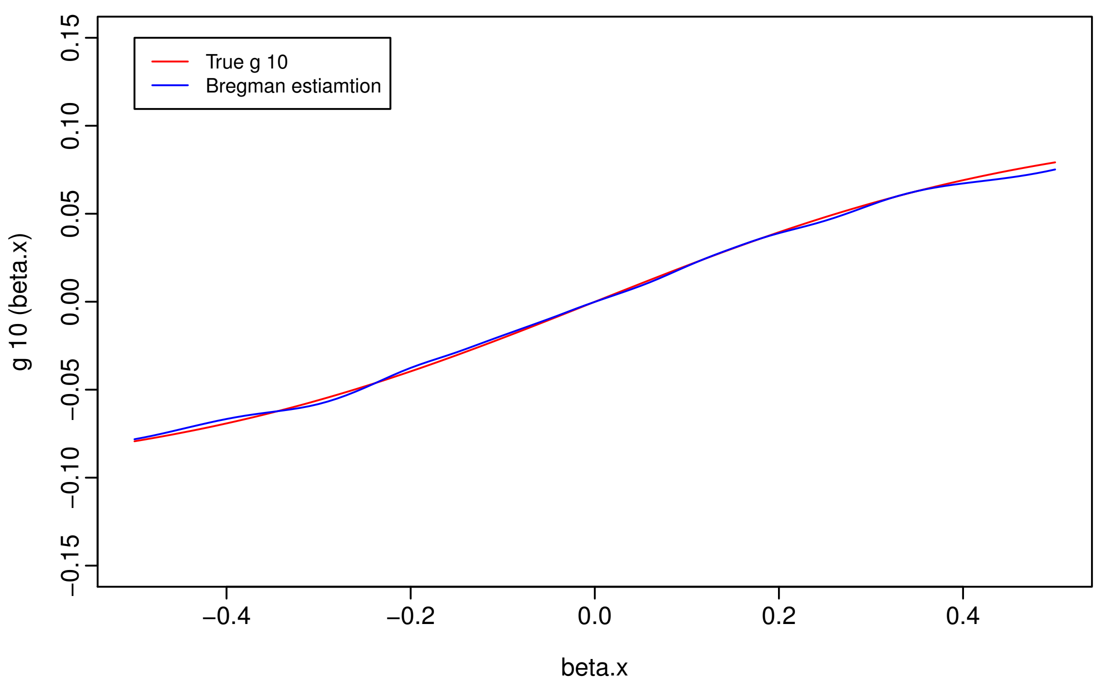


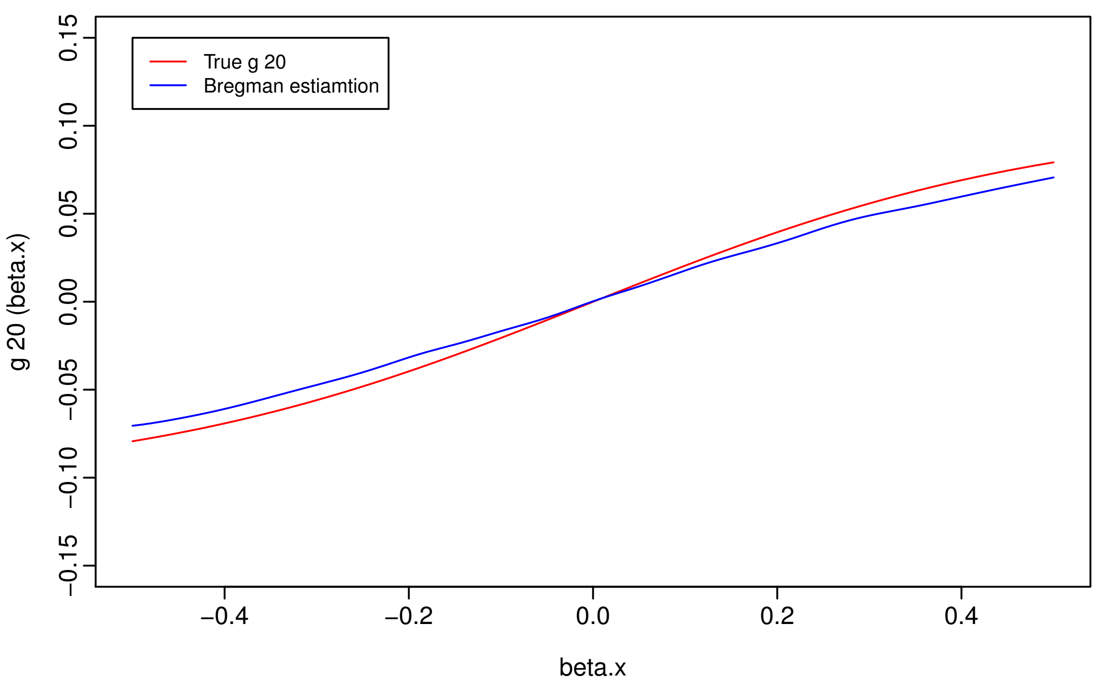
Function Recovery: Exponential



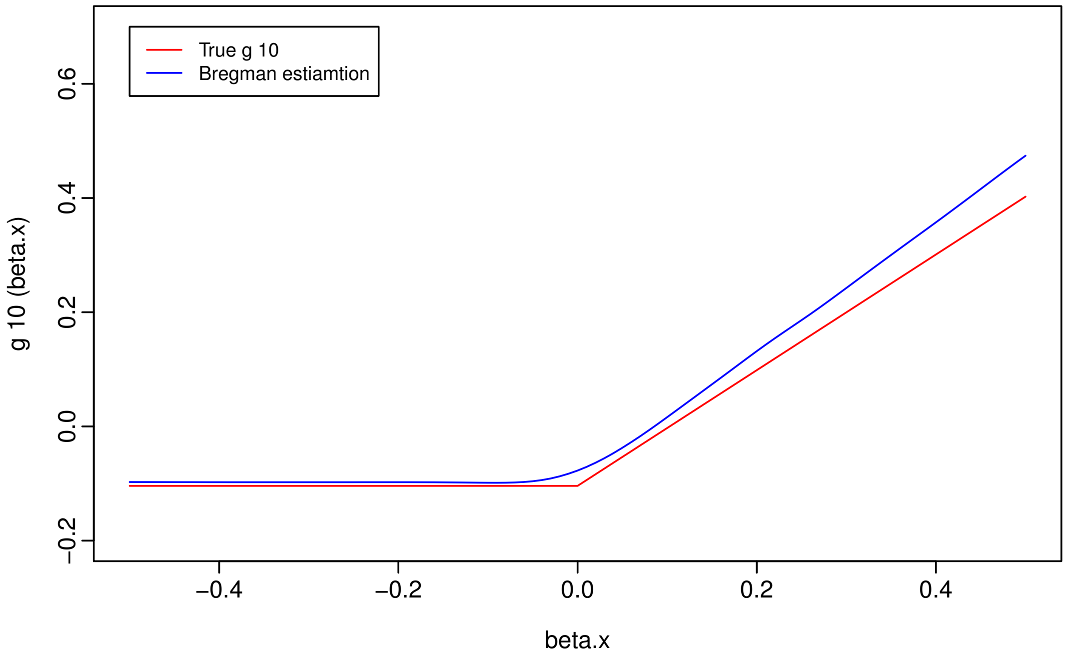


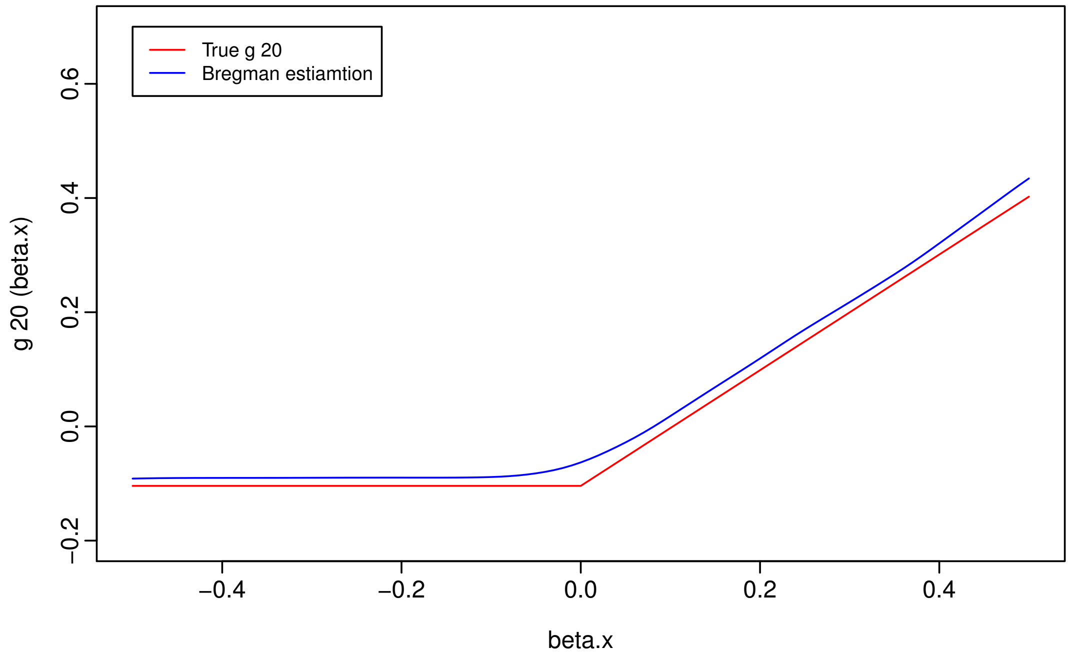
Function Recovery: Sigmoidal





Function Recovery: Rectifying





Summary

- Multiple Index Models provide a natural semi-parametric framework in many settings: in neural coding in particular

- Their use till now has been limited due to problems with inference given non-convex objectives

- We provide a surrogate loss that is convex in the projection weights

- Modern non-parametrics needs to marry recent advances in convex/variational optimization and structural constraints to classical non-parametrics

Thank You!