An Algorithmic Approach to Nonparametric Convex Regression

Rahul Mazumder

Massachussets Institute of Technology

June, 2016

Nonparametric Function Estimation

- Data $(y_i, x_i), i = 1, \ldots, n$. Response: y, covariate $x \in \Re^d$.
- Approximate the "data generating" mechanism:

$$y = \underbrace{\psi(x)}_{\text{Unknown}} \quad + \quad \underbrace{\epsilon}_{\text{Error}}$$

- Usual linear model is not flexible enough. Need more flexibility.
- Some popular examples in (Statistics/Machine Learning):
  - Smoothing methods
  - CART/Regression trees/Kernel SVMs/ Ensemble methods
  - Empirical Likelihood
  - Shape constraints on ψ
    (convexity/concavity, monotonicity, Lipschitz,...)

Nonparametric Function Estimation

- Data $(y_i, x_i), i = 1, \ldots, n$. Response: y, covariate $x \in \Re^d$.
- Approximate the "data generating" mechanism:

$$y = \underbrace{\psi(x)}_{\text{Unknown}} + \underbrace{\epsilon}_{\text{Error}}$$

- Usual linear model is not flexible enough. Need more flexibility.
- Some popular examples in (Statistics/Machine Learning):
  - Smoothing methods
  - CART/Regression trees/Kernel SVMs/ Ensemble methods
  - Empirical Likelihood
  - Shape constraints on ψ
    (convexity/concavity, monotonicity, Lipschitz,...)

A Computational Framework for Multivariate Convex Regression and its Variants

(Mazumder, Choudhury, Iyengar, Sen (2015) [preprint], `http://arxiv.org/pdf/1509.08165v1`)

Multivariate Convex Function Estimation

- Estimate $\psi : \Re^d \mapsto \Re$ such that it is convex

  Definition:

  $$\psi(\alpha x + (1-\alpha)x') \leq \alpha\psi(x) + (1-\alpha)\psi(x'),\ \forall\ x, x' \in \Re^d, \alpha \in [0,1]$$

- This leads to the natural least squares problem:

$$\hat{\psi} \in \underset{\psi \text{ is convex}}{\operatorname{argmin}} \quad \sum_{i=1}^{n} (y_i - \psi(x_i))^2, \tag{1}$$

- An appealing feature: no tuning parameters (e.g., choice of bandwidths as in smoothing methods)...

Multivariate Convex Function Estimation

- Lots of recent work in the area of shape constrained estimation
  — Cule et al. '10 and Seregin and Wellner '10 (density estimation)
  — Seijo and Sen '11; Glynn and Lim '12; Hannah and Dunson '13; Xu, Chen, Laferty '16, .. (regression function estimation)
- Applications in economics, operations research, reinforcement learning, others...
- Personal interests: Oceanography, Sports Analytics,...

Multivariate Convex Function Estimation

- Problem (1) is an infinite dimensional optimization problem (space of all convex functions in $\Re^d$)
- Can be reduced to a finite dimensional problem
- Why?
  Recall (equivalent) definitions of convexity of ψ:
  (a) $\psi(\alpha x + (1-\alpha)x') \leq \alpha\psi(x) + (1-\alpha)\psi(x')$ for $\alpha \in [0,1]$, $\forall x, x'$
  (b) $\exists \partial\psi(x')$ such that $\psi(x) \geq \psi(x') + \langle \partial\psi(x'), x - x' \rangle$, $\forall x, x'$
  (c) $\exists \partial\psi(x), \partial\psi(x')$ such that $\langle \partial\psi(x) - \partial\psi(x'), x - x' \rangle \geq 0$, $\forall x, x'$

  [$\partial\psi(x)$ is a subgradient of a convex function]

Multivariate Convex Function Estimation

- Problem (1) is an infinite dimensional optimization problem (space of all convex functions in $\Re^d$)
- Can be reduced to a finite dimensional problem
- Why?
  Recall (equivalent) definitions of convexity of ψ:
  (a) $\psi(\alpha x + (1-\alpha)x') \leq \alpha\psi(x) + (1-\alpha)\psi(x')$ for $\alpha \in [0,1]$, $\forall x, x'$
  (b) $\exists \partial\psi(x')$ such that $\psi(x) \geq \psi(x') + \langle \partial\psi(x'), x - x' \rangle$, $\forall x, x'$
  (c) $\exists \partial\psi(x), \partial\psi(x')$ such that $\langle \partial\psi(x) - \partial\psi(x'), x - x' \rangle \geq 0$, $\forall x, x'$

  [$\partial\psi(x)$ is a subgradient of a convex function]

Multivariate Convex Function Estimation

- Note that:

$$\hat{\psi} \in \operatorname{argmin} \sum_{i=1}^{n} (y_i - \psi(x_i))^2 \quad \text{s.t.} \;\; \psi \text{ is convex}$$

is equivalent to the Quadratic Program (QP):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta_i)^2 \\ \text{s.t.} \quad & \theta_j + \langle x_i - x_j, \xi_j \rangle \le \theta_i; \quad i \neq j \in \{1, \dots, n\}, \end{aligned} \tag{2}$$

- Estimates function values and subgradients at n different points
- Optimization variables:
  - $\theta_i \in \Re$ is function value at x_i for $i = 1, \dots, n$.
  - $\xi_i \in \Re^d$ is subgradient of ψ at x_i (that is: $\partial \psi(x_i)$) for $i = 1, \dots, n$.

Multivariate Convex Function Estimation

- The QP estimates $\theta_i = \psi(x_i)$ and $\xi_i = \partial\psi(x_i)$ for all $i = 1, \ldots, n$.
- How to extend to a function defined on all of $\Re^d$?
  (Only the convex hull: Conv$(x_1, \ldots, x_n)$ is statistically meaningful)

Multivariate Convex Function Estimation

- The QP estimates $\theta_i = \psi(x_i)$ and $\xi_i = \partial\psi(x_i)$ for all $i = 1, \ldots, n$.
- How to extend to a function defined on all of $\Re^d$?
  (Only the convex hull: Conv$(x_1, \ldots, x_n)$ is statistically meaningful)
- A natural interpolation scheme for $\hat{\psi}$:

$$\hat{\psi}(x) = \max_{j=1,\ldots,n} \left\{ \hat{\theta}_j + \langle x - x_j, \hat{\xi}_j \rangle \right\}$$

  leads to a convex function defined on $\Re^d$.
- ($\Longrightarrow$) the equivalence between Problem (2) and (1).

Computation?

- Convex regression can be solved with a QP $\Longrightarrow$ good in theory
- Question: How fast are off-the-shelf solvers, in practice?

Computation?

- Convex regression can be solved with a QP $\implies$ good in theory
- Question: How fast are off-the-shelf solvers, in practice?

n	d	Time (in secs) (SDTP3, cvx)	Time (in secs) MOSEK	Time (in secs) Our Algorithm
100	5	33	6	< 2
200	5	159	125	< 5
300	5	562	342	8
400	5	1640	1151	15
500	5	3745	4071	20

Table showing timings (in seconds) for solving the convex regression QP for a problem with n samples in d dimensions.

Computation?

Computational Considerations for Problem (2):

- Problem has $O(n^2)$ constraints, and $O(nd)$ variables.

- Off-the-shelf interior point methods (e.g. `cvx`):
  — cost at least $O(n^3 d^3)$
  — do not scale well for $n \geq 300$

- Desirable to develop tailor-made algorithms that:

  - **scale well**
    — Fast/reliable/accurate solutions for large problem sizes.

  - **are flexible**
    — Shape constraints (some coordinates non-negative, ↑, ↓, etc)
    — Constraints on the subgradients (Lipschitz, bounded, etc..)

An Algorithmic Framework

Write Problem (2) as:

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_i)^2 \\
\text{s.t.} \quad & \eta_{ij} = \theta_j + \langle \Delta_{ij}, \xi_j \rangle - \theta_i; \qquad i \neq j = 1, \ldots, n, \\
& \eta_{ij} \leq 0; \qquad i \neq j = 1, \ldots, n,
\end{aligned}
\tag{3}
$$

where, $\Delta_{ij} := x_i - x_j$ for all i, j.

Algorithmic Framework based on ADMM[1]

Define the Augmented Lagrangian corresponding to the above formulation as

$$\begin{aligned}\mathcal{L}_\rho((\xi_1,\ldots,\xi_n;\theta;\eta);\nu) \quad := \quad & \frac{1}{2}\sum_{i=1}^{n}(y_i-\theta_i)^2 \\ & + \sum_{i,j}\nu_{ij}\left(\eta_{ij}-(\theta_j+\langle\Delta_{ij},\xi_j\rangle-\theta_i)\right) \\ & + \quad \frac{\rho}{2}\sum_{i,j}\left(\eta_{ij}-(\theta_j+\langle\Delta_{ij},\xi_j\rangle-\theta_i)\right)^2\end{aligned}$$

where $\nu \in \Re^{n\times n}$ is the matrix of dual variables.

[1] Alternating Direction Method of Multipliers [Boyd, et al. '11; Bertsekas '99.]

MultiBlock ADMM: Algorithm 1

Initialize variables $(\xi_1^{(1)}, \ldots, \xi_n^{(1)})$, $\theta^{(1)}$, $\eta^{(1)}$ and $\nu^{(1)}$.
Perform the following Steps 1—4 for $k \geq 1$ till convergence.

1. Update the subgradients $(\xi_1, \ldots, \xi_n)$:

$$(\xi_1^{(k+1)}, \ldots, \xi_n^{(k+1)}) \in \underset{\xi_1, \ldots, \xi_n}{\operatorname{argmin}} \mathcal{L}_\rho \left((\xi_1, \ldots, \xi_n; \theta^{(k)}; \eta^{(k)}); \nu^{(k)} \right). \tag{4}$$

2. Update the function values θ:

$$\theta^{(k+1)} \in \underset{\theta}{\operatorname{argmin}} \mathcal{L}_\rho \left((\xi_1^{(k+1)}, \ldots, \xi_n^{(k+1)}; \theta; \eta^{(k)}); \nu^{(k)} \right). \tag{5}$$

3. Update the residual matrix η:

$$\eta^{(k+1)} \in \underset{\eta\,:\,\eta_{ij} \leq 0,\ \forall i,j}{\operatorname{argmin}} \mathcal{L}_\rho \left((\xi_1^{(k+1)}, \ldots, \xi_n^{(k+1)}; \theta^{(k+1)}; \eta); \nu^{(k)} \right). \tag{6}$$

4. Update the dual variable:

$$\nu_{ij}^{(k+1)} \leftarrow \nu_{ij}^{(k)} + \rho \left(\eta_{ij}^{(k+1)} - \left(\theta_j^{(k+1)} + \langle \Delta_{ij}, \xi_j^{(k+1)} \rangle - \theta_i^{(k+1)} \right) \right); \tag{7}$$

for $i, j = 1, \ldots, n$.

Update details

Updating subgradients: solving Problem (4)

- Compute:
$$\hat{\xi}_j = \Big(\sum_i \Delta_{ij}\Delta_{ij}^\top\Big)^{-1}\Big(\sum_i \Delta_{ij}\bar{\eta}_{ij}\Big)$$
where $\bar{\eta}_{ij} = \nu_{ij}/\rho + \eta_{ij} - (\theta_j - \theta_i)$.
- $\overline{\Delta}_j := \big(\sum_i \Delta_{ij}\Delta_{ij}^\top\big)^{-1}$ for $j = 1, \ldots, n$ can be computed offline
- With careful book-keeping: for $d \ll n$, the cost per iteration is $O(n^2)$.

Update details

Updating the function values: solving Problem (5)

- Reduces to solving the system:

$$(I + \rho D^\top D)\hat{\theta} = \underbrace{Y + D^\top \mathrm{vec}(\nu) + \rho D^\top \mathrm{vec}(\tilde{\eta})}_{:=v}. \tag{8}$$

- A direct inversion to solve for θ will have a complexity of $O(n^3)$.

Update details

Updating the function values: solving Problem (5)

- Reduces to solving the system:

$$(I + \rho D^\top D)\hat{\theta} = \underbrace{Y + D^\top \text{vec}(\nu) + \rho D^\top \text{vec}(\tilde{\eta})}_{:=v}. \tag{8}$$

- A direct inversion to solve for θ will have a complexity of $O(n^3)$.
- Exploit structure of D:

$$(I + \rho D^\top D) = (1 + 2n\rho)I - 2\rho 11^\top,$$

  Compute $(I + \rho D^\top D)^{-1}$ in $O(n)$ flops, given v.

Update details

- Updating the residuals: solving Problem (6), is simple.

- The cost per iteration of Algorithm 1 is $O(\max\{n^2 d, nd^3\})$, with an additional $O(n^2 d^2 + nd^3)$ for the offline computation of matrix inverses

- Overall cost per iteration is $O(n^2)$ for $d \ll n$.

Caveats and Alternatives

- Multiblock ADMM (Algorithm 1) has limited (theoretical) convergence guarantees
  (Chen et al. '14)

- Modified version: Algorithm 2 has convergence guarantees.
  In particular: $O(\frac{1}{\delta})$ many iterations to get an δ-accurate solution

- Practically Algorithms 1 and 2 are often similar (Algorithm 2 may be marginally slower)

Algorithm in action

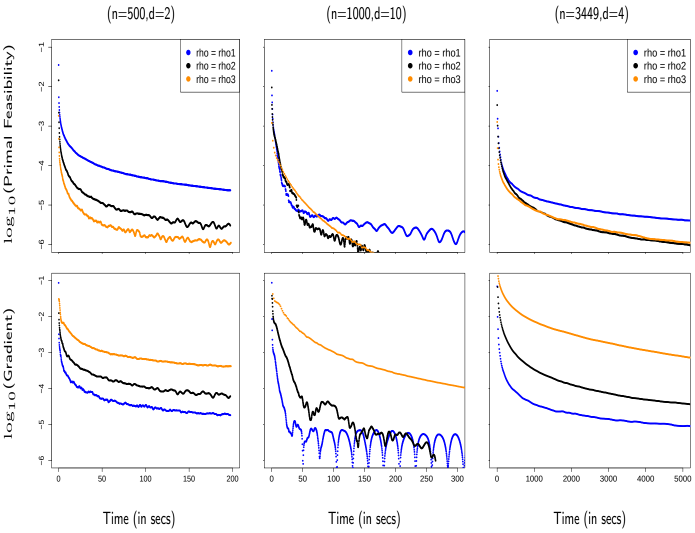

Figure: Algorithm 1 with time, for three different examples. Three different ρ values, denoted by 'rho1', 'rho2', 'rho3', were taken to be $0.1/n, 1/n, 10/n$ respectively.

Smooth convex function estimates?

- Recall, interpolant is given by:

$$\hat{\psi}(x) = \max_{j=1,\dots,n} \left\{ \hat{\theta}_j + \langle x - x_j, \hat{\xi}_j \rangle \right\}.$$

- $\hat{\psi}(x)$ is not smooth in x.
- Is it possible to obtain $\hat{\psi}(x)$ that is both **convex and smooth** in x?
- Smoothness is traditionally imposed via some form of "averaging" wrt to a kernel. Smoothness and shape constraints together are typically hard to achieve.
- Our approach: use a technique presented in *"Smooth minimization of nonsmooth functions"* by Nesterov '05, *Math. Programming.*

Smooth convex function estimates?

- Note that
$$\hat{\psi}(x) = \max\left\{a_1^\top x + b_1, \ldots, a_m^\top x + b_m\right\}.$$
- Observe that $\hat{\psi}$ admits:
$$\begin{aligned}\hat{\psi}(x) = \quad \max_{w} \quad & \sum_{i=1}^{m} w_i\left(a_i^\top x + b_i\right) \\ \text{s.t.} \quad & \sum_{i=1}^{m} w_i = 1, w_i \geq 0, i = 1, \ldots, m,\end{aligned}$$
- Why is $x \mapsto \hat{\psi}(x)$ non-differentiable? How can it be "fixed"?

Smooth convex function estimates?

- Note that
$$\hat{\psi}(x) = \max\left\{a_1^\top x + b_1, \ldots, a_m^\top x + b_m\right\}.$$

- Observe that $\hat{\psi}$ admits:

$$\begin{aligned}\hat{\psi}(x) = \quad & \max_w \quad \sum_{i=1}^m w_i\left(a_i^\top x + b_i\right) \\ & \text{s.t.} \quad \sum_{i=1}^m w_i = 1, w_i \geq 0, i = 1, \ldots, m,\end{aligned}$$

- Why is $x \mapsto \hat{\psi}(x)$ non-differentiable? How can it be "fixed"?
- Consider the following perturbed version:

$$\begin{aligned}\tilde{\psi}(x;\tau) = \quad & \max_w \quad \sum_{i=1}^m w_i\left(a_i^\top x + b_i\right) - \tau\|w - 1/m\|_2^2 \\ & \text{s.t.} \quad \sum_{i=1}^m w_i = 1, w_i \geq 0, i = 1, \ldots, m,\end{aligned}$$

Smooth convex function estimates?

What are the properties of $\tilde{\psi}(x;\tau)$?

- $\tilde{\psi}(x;\tau)$ is convex in x
- $\tilde{\psi}(x;\tau)$ is an $O(\tau)$ uniform approximation to $\tilde{\psi}(x;0) := \hat{\psi}(x)$.

$$\hat{\psi}(x) - \tau \sup_{w \in Q} \|w - 1/m\|_2^2 \leq \tilde{\psi}(x;\tau) \leq \hat{\psi}(x)$$

- Also:

$$\|\nabla\tilde{\psi}(x_1;\tau) - \nabla\tilde{\psi}(x_2;\tau)\| \leq \frac{\lambda_{\max}(A^\top A)}{\tau}\|x_1 - x_2\|$$

Thus: $x \mapsto \tilde{\psi}(x;\tau)$ has gradient Lipschitz continuous with parameter $O(1/\tau)$.

- Is the choice $\|w - 1/m\|_2^2$ special?

- Is the choice $\|w - 1/m\|_2^2$ special?

  NO. Other smooth approximations possible.

- Is the choice $\|w - 1/m\|_2^2$ special?
  NO. Other smooth approximations possible.
- If Q is the simplex in $\Re^m$ and $\rho(\cdot)$ a proximity (prox) function of Q, i.e.,
  - $\rho(\cdot)$ is continuously differentiable
  - $\rho(\cdot)$ is strongly convex on Q (wrt norm $\|\cdot\|_\dagger$)
- The following is a uniform, convex, smooth approximation of $\hat{\psi}(x)$

$$\begin{aligned} \tilde{\psi}_\rho(x;\tau) = \quad & \max_w \quad \sum_{i=1}^m w_i \left(a_i^\top x + b_i\right) - \tau\rho(w) \\ & \text{s.t.} \quad \sum_{i=1}^m w_i = 1, w_i \geq 0, i = 1, \ldots, m, \end{aligned}$$

Smoothing in action

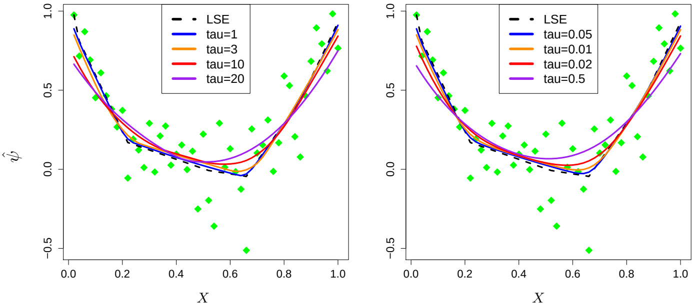

Figure: Plots of the data points and the convex LSE $\hat{\psi}$ with the bias corrected smoothed estimators for four different choices to τ using the squared error prox function (left panel) and entropy prox function (right panel).

Lipschitz Convex Regression

- The convex LSE described in (2) suffers from over-fitting, especially near the boundary of the convex hull of the design points x_i's.
- The norms of the fitted subgradients $\hat{\xi}_i$'s near the boundary can become arbitrarily large
- A remedy to this over-fitting: consider LS minimization over the class of convex functions that are uniformly Lipschitz with a known bound.

$$C_L := \left\{ \psi : \mathfrak{X} \to \Re \mid \psi \text{ is convex}, \sup_{x \in \mathfrak{X}} \|\partial\psi(x)\| \leq L \right\}.$$

Lipschitz Convex Regression

- Let $\hat{\psi}_L$ denote the LSE when minimizing the SSE over the class C_L, i.e.,

$$\hat{\psi}_L \in \text{argmin} \sum_{i=1}^{n}(y_i - \psi(x_i))^2 \quad \text{s.t.} \quad \psi \in C_L$$

- Solution to above problem can be obtained by solving:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|Y - \theta\|_2^2 \\ \text{s.t.} \quad & \theta_j + \langle x_i - x_j, \xi_j \rangle \leq \theta_i;\ i \neq j = 1, \ldots, n; \\ & \|\xi_j\| \leq L, \quad j = 1, \ldots, n. \end{aligned}$$

- For example, $\|\cdot\| \in \{\|\cdot\|_2, \|\cdot\|_1, \|\cdot\|_\infty\}$.

Lipschitz Convex Regression

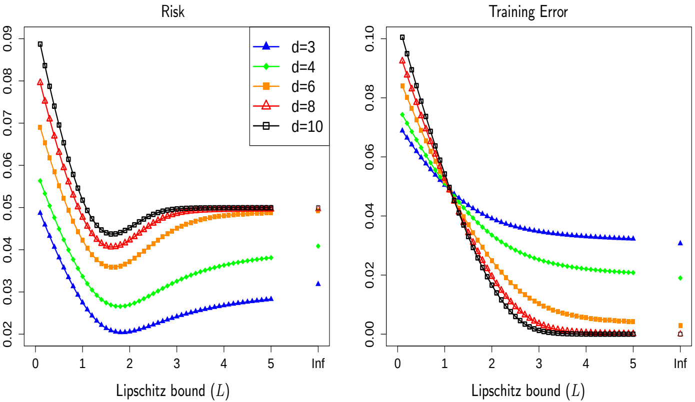

Figure: [Left panel]: the simulated risk of the Lipschitz convex estimator as the Lipschitz bound L varies (L $=$ Inf gives the usual convex LSE) for 5 different dimension values (d). [Right panel]: the training error as the Lipschitz bound L varies, for the same examples appearing in the left panel.

Flexible Convex Regression

- Lipschitz convex regression:
  - Computation?

Flexible Convex Regression

- Lipschitz convex regression:
  - Computation?
    Slightly harder, but not much. Same framework applies.

Flexible Convex Regression

- Lipschitz convex regression:
  - Computation?
    Slightly harder, but not much. Same framework applies.
  - Does the smoothing method work?

Flexible Convex Regression

- Lipschitz convex regression:
  - Computation?
    Slightly harder, but not much. Same framework applies.
  - Does the smoothing method work?
    Yes.

Flexible Convex Regression

- Lipschitz convex regression:
  - Computation?
    Slightly harder, but not much. Same framework applies.
  - Does the smoothing method work?
    Yes.
- What if $\psi(x)$ is (partially) increasing in coordinate x_1?

Flexible Convex Regression

- Lipschitz convex regression:
  - Computation?
    Slightly harder, but not much. Same framework applies.
  - Does the smoothing method work?
    Yes.
- What if $\psi(x)$ is (partially) increasing in coordinate x_1?
  Add constraint $\xi_1 \geq 0$ to problem.

Statistical Property

Theorem (**M.**, Choudhury, Iyengar, Sen '15)
Consider observations $(y_i, x_i), i = 1, \dots, n$ such that

$$y_i = \psi(x_i) + \epsilon_i,$$

where $\psi : \Re^d \to \Re$ is an unknown convex function (d is fixed). We assume that

(i) the support of x is $\mathfrak{X} = [0,1]^d$

(ii) $\psi \in C_{L_0}$ for some $L_0 > 0$

(iii) the $x_i \in \mathfrak{X}$'s are fixed constants and

(iv) ϵ_i's are independent mean zero sub-Gaussian errors.

We have for any $L > L_0$,

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{\psi}_{n,L}(x_i) - \psi(x_i))^2 = O_P(r_n),$$

where

$$r_n = \begin{cases} n^{-2/(d+4)} & \text{if } d = 1,2,3, \\ n^{-1/4}(\log n)^{1/2} & \text{if } d = 4, \\ n^{-1/d} & \text{if } d \geq 5. \end{cases}$$

Sparse Convex Regression

- Multivariate convex regression is statistically troublesome, when:
  — n, d are comparable
  — d is large
  — curse of dimensionality kicks in

- Some form of dimension reduction is required: **Sparsity?**

- $\psi(x) : \Re^d \mapsto \Re$ is a convex function, that depends upon an (unknown) subset of $k \ll d$ variables.

$$\psi(x_1, \ldots, x_d) = g(x_{i_1}, \ldots, x_{i_k}),\ g \text{ convex and } \underbrace{\{i_1, \ldots, i_k\}}_{\text{Unknown}} \subset \{1, \ldots, d\}.$$

Denote the above collection of functions by $\mathcal{F}_k$

Variable Selection in Multivariate Convex Regression with Discrete Optimization

(Mazumder (2016) [work in progress])

Sparse Convex Regression

- Usual convex regression:

$$\min \quad \sum_{i=1}^{n} |y_i - \psi(x_i)|^q \quad \text{s.t.} \quad \psi \text{ is convex}$$

  for $q \in \{1, 2\}$.
- Sparse convex regression:

$$\min \quad \sum_{i=1}^{n} |y_i - \psi(x_i)|^q \quad \text{s.t.} \quad \psi \in \mathcal{F}_k.$$

Sparse Convex Regression

- Usual convex regression:

$$\min \quad \sum_{i=1}^{n} |y_i - \psi(x_i)|^q \quad \text{s.t.} \quad \psi \text{ is convex}$$

  for $q \in \{1, 2\}$.

- Sparse convex regression:

$$\min \quad \sum_{i=1}^{n} |y_i - \psi(x_i)|^q \quad \text{s.t.} \quad \psi \in \mathcal{F}_k.$$

  is equivalent to:

$$\begin{array}{ll} \min & \sum_{i=1}^{n} |y_i - \theta_i|^q \\ \text{subject to} & \theta_j + \langle x_i - x_j, \xi_j \rangle \leq \theta_i, \quad i \neq j \in \{1, \ldots, n\}, \\ & \sum_{i=1}^{d} \mathbf{1}(\xi^i \neq 0) \leq k, \end{array}$$

Sparse Convex Regression

- Caveat: this is a combinatorial optimization problem (possibly NP hard)
- Special instance of this problem:

$$\psi(x) = x^\top \beta \qquad \text{(Sparse/Variable Selection in Linear Regression)}$$

- Tools described before for Convex LS regression do not apply here.
- New approach is necessary. We use modern discrete optimization methods.

Sparse Convex Regression

- Can be expressed as a Mixed Integer Quadratic Optimization (MIO) Problem

- A general form of MIO is representable as:

$$\begin{array}{ll} \underset{\boldsymbol{\alpha}}{\text{minimize}} & \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{a} \\ \text{subject to} & \mathbf{A}\boldsymbol{\alpha} \leq \mathbf{b} \\ & \alpha_i \in \{0,1\}, \quad \forall i \in \mathcal{I} \\ & \alpha_j \in \mathbb{R}_+, \quad \forall j \notin \mathcal{I}, \end{array}$$

  $\mathbf{a} \in \Re^m, \mathbf{A} \in \Re^{k \times m}, \mathbf{b} \in \Re^k$ and $\mathbf{Q} \in \Re^{m \times m}$ (PSD) problem-parameters.

- Sparse convex regression:
  — $q = 1$ is a Mixed Integer Linear Program
  — $q = 1$ is a Mixed Integer Quadratic Program

Sparse Convex Regression

- Huge improvements in Algorithms & Software over past 25+ years
- Algorithms speed-up: 780,000 times
- Hardware speed-up: 570,000 times
- Total speed-up: **450 Billion times!**
  (As of May, 2016 this is **850 billion!**)
- Solve (with certificates) practical sized problems in times relevant for applications considered
- Successfully used across wide range of applications in Operations Research

Sparse Convex Regression

- Sparse Convex Regression admits a MIO representation, with:
  - d binary variables
  - $O(nd)$ continuous variables
  - $O(n^2)$ linear inequalities

- In spite of progress in MIO, this problem is challenging solve for large instances.

- New algorithmic tools are required for scalability:
  - Constraint generation, Cutting plane methods (Nemhauser, Wolsey '99)
  - Outer approximation methods, exploiting separability of loss function (Hijazi, et. al. '13; Vielma, et al '15)

- Competing method: Xu, Chen and Lafferty '16 (AC/DC) method

- AC/DC method requires the covariates to be independent (+ other regularity conditions) to identify right variables

- Preliminary findings:
  - Discrete optimization method makes better variable identification (by 10-30% better) for $n < d \approx 100$.
  - AC/DC method requires larger n than Discrete Optimization method, to identify all active variables.

Summary

- Many challenging and deep algorithmic questions in shape restricted estimation (generally nonparametric function estimation)

- A rigorous optimization lens often leads to newer perspectives and complements our statistical understanding

- Nonparametric function estimation $\longleftrightarrow$ Mathematical Programming

Thanks for your attention!