Regularised Additive Least Squares Regression

Calvin & Kirthevasan

May 7, 2015

Nonparametric Regression

Given data: $(X_i, Y_i)_{i=1}^n \quad$ where $(X_i, Y_i) \sim P_{XY}$
Estimate $f(x) = \mathbb{E}[Y|X = x]$.

Nonparametric Regression

- Assume only smoothness of f. No parametric form assumed.
- E.g.: Nadaraya-Watson, Support Vector Regression, Locally Polynomial Regression, Splines etc.

Nonparametric Regression

Given data: $(X_i, Y_i)_{i=1}^n$ where $(X_i, Y_i) \sim P_{XY}$
Estimate $f(x) = \mathbb{E}[Y|X = x]$.

Nonparametric Regression

- Assume only smoothness of f. No parametric form assumed.
- E.g.: Nadaraya-Watson, Support Vector Regression, Locally Polynomial Regression, Splines etc.

- **Kernel Ridge Regression**
  Use a kernel k and its associated RKHS $\mathcal{H}_k$,

$$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{H}_k} \sum_{i=1}^{n} (Y_i - f(X_i))^2 + \lambda \|f\|_{\mathcal{H}_k}^2$$

The Bane of Nonparametric Methods

- The curse of dimensionality: Sample complexity is exponential in D. Typically work well only under $4 - 6$ dimensions.
- Difficult to identify/ exploit structure in the problem.

This work: Address above via additive estimate for f,

$$\hat{f}(\cdot) = \hat{f}^{(1)}(\cdot) + \hat{f}^{(2)}(\cdot) + \ldots + \hat{f}^{(M)}(\cdot)$$

Outline

- Additive Least Squares Regression
- High dimensional nonparametric regression
  - “Statistically” simpler structures.
- Function selection
- Implementation
  - Comparison of optimisation procedures.

Additive Kernel Regression

Recall,

$$\hat{f}(\cdot) = \hat{f}^{(1)}(\cdot) + \hat{f}^{(2)}(\cdot) + \ldots + \hat{f}^{(M)}(\cdot)$$

Additive Kernel Regression

Recall,

$$\hat{f}(\cdot) = \hat{f}^{(1)}(\cdot) + \hat{f}^{(2)}(\cdot) + \ldots + \hat{f}^{(M)}(\cdot)$$

Given: kernels $k^{(j)}$ and the RKHS $\mathcal{H}_{k^{(j)}}$ for each $\hat{f}^{(j)}$.
Optimise over $f^{(j)} \in \mathcal{H}_{k^{(j)}}$, $j = 1, \ldots, M$

Additive Kernel Regression

Recall,

$$\hat{f}(\cdot) \;=\; \hat{f}^{(1)}(\cdot) \;+\; \hat{f}^{(2)}(\cdot) \;+\; \dots \;+\; \hat{f}^{(M)}(\cdot)$$

Given: kernels $k^{(j)}$ and the RKHS $\mathcal{H}_{k^{(j)}}$ for each $\hat{f}^{(j)}$.
Optimise over $f^{(j)} \in \mathcal{H}_{k^{(j)}}$, $j = 1, \dots, M$

$$\{\hat{f}^{(j)}\}_{j=1}^{M} = \operatorname*{argmin}_{f^{(j)} \in \mathcal{H}_{k^{(j)}}, j=1,\dots,M} F\left(\{f^{(j)}\}_{j=1}^{M}\right)$$

$$F\left(\{f^{(j)}\}_{j=1}^{M}\right) \;= \frac{1}{2}\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{M} f^{(j)}(X_i)\right)^2 + \lambda \sum_{j=1}^{M} \|f^{(j)}\|_{\mathcal{H}_{k^{(j)}}}$$

Additive Kernel Regression

Representer Theorem: $\hat{f}^{(j)}(\cdot) = \sum_{i=1}^{n} \alpha_i^{(j)} k^{(j)}(\cdot, X_i)$.

Additive Kernel Regression

Representer Theorem: $\hat{f}^{(j)}(\cdot) = \sum_{i=1}^{n} \alpha_i^{(j)} k^{(j)}(\cdot, X_i)$.

Write $\alpha^{(j)} \in \mathbb{R}^n$, $\boldsymbol{\alpha} \in \mathbb{R}^{nM}$. The objective reduces to

$$F_1(\boldsymbol{\alpha}) = \frac{1}{2}\left\| Y - \sum_{j=1}^{m} K^{(j)}\alpha^{(j)} \right\|_2^2 + \lambda \sum_{j=1}^{M} \sqrt{\alpha^{(j)\top} K^{(j)} \alpha^{(j)}}.$$

This is convex.

Higher Dimensional Regression

$$\hat{f}(\cdot) \;=\; \hat{f}^{(1)}(\cdot) \;+\; \hat{f}^{(2)}(\cdot) \;+\; \ldots \;+\; \hat{f}^{(M)}(\cdot)$$

Higher Dimensional Regression

$$\hat{f}(\cdot) \;=\; \hat{f}^{(1)}(\cdot) \;+\; \hat{f}^{(2)}(\cdot) \;+\; \ldots \;+\; \hat{f}^{(M)}(\cdot)$$

Idea: Choose $k^{(j)}$ to be "simple".
The sum $\hat{f}$ will still be "simpler" than estimating on a full Kernel.

Higher Dimensional Regression

$$\hat{f}(\cdot) \;=\; \hat{f}^{(1)}(\cdot) \;+\; \hat{f}^{(2)}(\cdot) \;+\; \dots \;+\; \hat{f}^{(M)}(\cdot)$$

Idea: Choose $k^{(j)}$ to be "simple".
The sum $\hat{f}$ will still be "simpler" than estimating on a full Kernel.

Full Kernel

$$k(x, x') = \exp\left(\frac{\|x - x'\|^2}{2h^2}\right) = \prod_{j=d}^{D} \exp\left(\frac{(x_d - x'_d)^2}{2h^2}\right) = \prod_{d=1}^{D} k_d(x_d, x'_d)$$

Higher Dimensional Regression

$$\hat{f}(\cdot) \;=\; \hat{f}^{(1)}(\cdot) \;+\; \hat{f}^{(2)}(\cdot) \;+\; \ldots \;+\; \hat{f}^{(M)}(\cdot)$$

Idea: Choose $k^{(j)}$ to be "simple".
The sum $\hat{f}$ will still be "simpler" than estimating on a full Kernel.

Full Kernel

$$k(x, x') = \exp\left(\frac{\|x - x'\|^2}{2h^2}\right) = \prod_{j=d}^{D} \exp\left(\frac{(x_d - x'_d)^2}{2h^2}\right) = \prod_{d=1}^{D} k_d(x_d, x'_d)$$

Why Simpler kernels ? More bias, but better variance

Higher Dimensional Regression - ESP Kernels

$$k^{(1)}(x, x') = \sum_{1 \le i \le D} k_i(x_i, x_i')$$

$$k^{(2)}(x, x') = \sum_{1 \le i_1 < i_2 \le D} k_{i_1}(x_{i_1}, x'_{i_1}) k_{i_2}(x_{i_2}, x'_{i_2})$$

$$k^{(M)}(x, x') = \sum_{1 \le i_1 < i_2 < \cdots < i_M \le D} \prod_{d=1}^{M} k_{i_d}(x_{i_d}, x'_{i_d})$$

Higher Dimensional Regression - ESP Kernels

$$k^{(1)}(x, x') = \sum_{1 \le i \le D} k_i(x_i, x'_i)$$

$$k^{(2)}(x, x') = \sum_{1 \le i_1 < i_2 \le D} k_{i_1}(x_{i_1}, x'_{i_1}) k_{i_2}(x_{i_2}, x'_{i_2})$$

$$k^{(M)}(x, x') = \sum_{1 \le i_1 < i_2 < \cdots < i_M \le D} \prod_{d=1}^{M} k_{i_d}(x_{i_d}, x'_{i_d})$$

- Combinatorially large number of terms.
- Observation: $k^{(j)}$ is the j^{th} elementary symmetric polynomial of base kernels k_i.
- Computable in $O(DM)$ time using Newton-Girard Formulae.

Higher Dimensional Regression

Results

Dataset (D, n)	Add-KR	KRR	NW	GP	SVR
Speech (21, 520)	**0.02269**	0.02777	0.11207	0.02531	0.22431
Music (90, 1000)	**0.91627**	0.91922	1.05745	0.94329	1.07009
Tele-motor (19, 300)	**0.06059**	0.06488	0.20119	0.06678	0.38038
Housing (12, 256)	**0.31285**	0.35947	0.42087	0.67566	1.15272
Blog (91, 700)	**1.43288**	1.53227	1.49305	1.64429	1.66705
Forest Fires (10, 210)	0.30675	0.32618	0.37199	**0.29038**	0.70154
Propulsion (15, 400)	0.04167	0.01396	0.11237	**0.00355**	0.74511

Also comparisons with k-NN, Locally Linear/Quadratic regression.

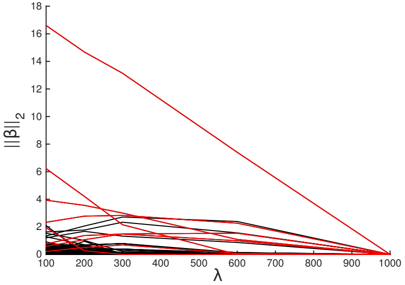
Function Selection

Important genomics task: Identify interdependent effect of mutations.

f has 50 variables, but true interactions are pairwise (or low order) and sparse.

$$f(x_1^{50}) = f(x_2, x_7) + f(x_{21}, x_{34}) + \cdots + f(x_{12}, x_{49})$$

4/50 individual effects and 4/1225 pairwise effects



Recovers all true nonzero functions with FPR=3.7%

Optimization Problem

Recall objective: $\alpha^{(j)} \in \mathbb{R}^n$, $\boldsymbol{\alpha} \in \mathbb{R}^{nM}$

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\left\| Y - \sum_{j=1}^{m} K^{(j)}\alpha^{(j)} \right\|_2^2 + \lambda \sum_{j=1}^{M} \sqrt{\alpha^{(j)\top} K^{(j)} \alpha^{(j)}}$$

- Challenge: generalized sum-of-norms regularization:

$$\sqrt{\alpha^{(j)\top} K^{(j)} \alpha^{(j)}} = \|R^{(j)}\alpha^{(j)}\|_2 \text{ where } K^{(j)} = R^{(j)\top} R^{(j)}$$

Optimization Problem

Recall objective: $\alpha^{(j)} \in \mathbb{R}^n$, $\boldsymbol{\alpha} \in \mathbb{R}^{nM}$

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\left\| Y - \sum_{j=1}^{m} K^{(j)}\alpha^{(j)} \right\|_2^2 + \lambda \sum_{j=1}^{M} \sqrt{\alpha^{(j)\top} K^{(j)} \alpha^{(j)}}$$

- Challenge: generalized sum-of-norms regularization:

$$\sqrt{\alpha^{(j)\top} K^{(j)} \alpha^{(j)}} = \|R^{(j)}\alpha^{(j)}\|_2 \text{ where } K^{(j)} = R^{(j)\top} R^{(j)}$$

Rewrite with Cholesky decomposition: $K^{(j)} = L^{(j)} L^{(j)\top}$.
Let $\beta^{(j)} = L^{(j)\top}\alpha^{(j)}$

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{nM}} \frac{1}{2}\left\| Y - \sum_{j=1}^{m} L^{(j)}\alpha^{(j)} \right\|_2^2 + \lambda \sum_{j=1}^{M} \|\beta^{(j)}\|_2$$

- Group lasso problem!

Optimization methods

- Subgradient / Proximal gradient: iteration cost $O(n^2 M)$
- ADMM: iteration cost $O(n^2 M^2)$
  Main cost: solve triangular $nM \times nM$ system in primal update

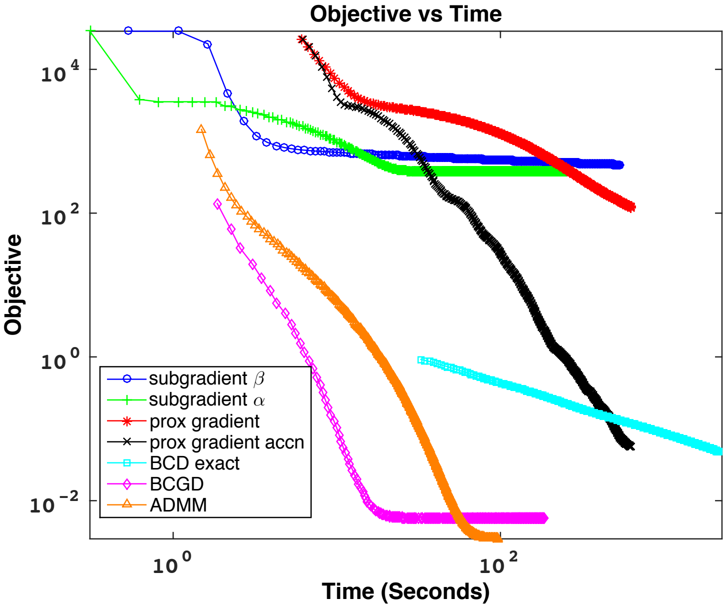
Optimization methods

- Subgradient / Proximal gradient: iteration cost $O(n^2M)$
- ADMM: iteration cost $O(n^2M^2)$
  Main cost: solve triangular $nM \times nM$ system in primal update
- Exact BCD: iteration cost $O(n^3M)$
  - Quickly solve 1d problem for Δ
    $\| - (\Delta L^{(j)}{}^\top L^{(j)} + \lambda I)^{-1}(L^{(j)}{}^\top(\sum_{i \neq j} L^{(i)}\beta^{(i)}))\| = 1$
  - Then solve $n \times n$ system for
    $\beta^{(j)} \leftarrow -(\Delta L^{(j)}{}^\top L^{(j)} + \lambda I)^{-1}(L^{(j)}{}^\top(\sum_{i \neq j} L^{(i)}\beta^{(i)}))/\Delta$

Optimization methods

- Subgradient / Proximal gradient: iteration cost $O(n^2M)$
- ADMM: iteration cost $O(n^2M^2)$
  Main cost: solve triangular $nM \times nM$ system in primal update
- Exact BCD: iteration cost $O(n^3M)$
  - Quickly solve 1d problem for Δ
    $\| -(\Delta {L^{(j)}}^\top L^{(j)} + \lambda I)^{-1}({L^{(j)}}^\top(\sum_{i\neq j} L^{(i)}\beta^{(i)}))\| = 1$
  - Then solve $n \times n$ system for
    $\beta^{(j)} \leftarrow -(\Delta {L^{(j)}}^\top L^{(j)} + \lambda I)^{-1}({L^{(j)}}^\top(\sum_{i\neq j} L^{(i)}\beta^{(i)}))/\Delta$
- BCGD: iteration cost $O((n^2 + nM)M)$
  - Block coordinate gradient descent (Tseng & Yun, 2007)
  - Diagonal Hessian approximation - closed form update
  - Skip backtracking since expensive

Optimization results



Thanks!