Sparse Nonparametric Regression
Using the Rodeo

John Lafferty
Computer Science Dept, and
Machine Learning Dept.
Carnegie Mellon University

Larry Wasserman
Dept. of Statistics, and
Machine Learning Dept.
Carnegie Mellon University
Rodeo: Regularization of derivative expectation operator

- A *general strategy* for nonparametric estimation: Regularize derivatives of estimator with respect to smoothing parameters
- A *simple new algorithm* for simultaneous bandwidth and variable selection in nonparametric regression
- *Theoretical analysis*: Algorithm correctly determines relevant variables, with high probability, and achieves (near) optimal minimax rate of convergence
- *Examples* showing performance consistent with theory
Outline

- Concepts from parametric and nonparametric regression
- Rodeo: Sparse nonparametric regression
- Rodeo for sparse nonparametric density estimation
- Briefly: Learning sparse graph structure from data
Two Background Ideas

- Variable (feature) selection
  - Lasso: least absolute shrinkage and selection operator
  - Key concept: sparsity
- Nonparametric regression (smoothing)

We are interested in the following question: how do we do variable selection in the model

\[ Y = m(x_1, \ldots, x_d) + \epsilon \]
Variable Selection in Linear Regression

\[ Y = \sum_{j=1}^{d} \beta_j X_j + \epsilon = X^T \beta + \epsilon \]

where \( d \) might be larger than \( n \). Predictive risk

\[ R = \mathbb{E}(Y_{new} - X_{new}^T \beta)^2. \]

Want to choose subset \( (X_j : j \in S), S \subset \{1, \ldots, d\} \) to make \( R \) small.

Bias-variance tradeoff:

small \( S \) \( \implies \) Bias \( \uparrow \) Variance \( \downarrow \)

large \( S \) \( \implies \) Bias \( \downarrow \) Variance \( \uparrow \)
Variable Selection in Linear Regression

Old methods:

1. **forward stepwise** (matching pursuit)
2. **ridge regression**

\[
\text{minimize } \sum_i (Y_i - X_i^T \hat{\beta}_j)^2 + \lambda \sum_j \beta_j^2
\]

Newer method: **Lasso** (basis pursuit):

\[
\text{minimize } \sum_i (Y_i - X_i^T \hat{\beta}_j)^2 + \lambda \sum_j |\beta_j|
\]
Lasso/Basis Pursuit

(Chen & Donoho, 1994; Tibshirani, 1996)

\[ \sum_{j=1}^{d} |\beta_j| \leq t \quad \text{Level sets of squared error} \]

For orthogonal designs, solution given by soft thresholding

\[ \hat{\beta}_j = \text{sign}(\beta_j) (|\beta_j| - \lambda)_+ \]
Oracle Inequality
(Donoho & Johnstone, 1994)

Let \( \lambda = \sigma \sqrt{2 \log d/n} \). Then, for every \( \beta \in \mathbb{R}^d \),

\[
\mathbb{E}_\beta \| \hat{\beta}_\lambda - \beta \|^2 \leq (2 \log d + 1) \left( \frac{\sigma^2}{n} + R_{\text{oracle}} \right)
\]

where \( R_{\text{oracle}} \) is the risk of an oracle that knows which variables to include.

No estimator can get substantially closer to the oracle in the sense that, as \( n \to \infty \),

\[
\inf_{\hat{\beta}} \sup_{\beta \in \mathbb{R}^n} \frac{\mathbb{E}_\beta \| \hat{\beta} - \beta \|^2}{\sigma^2/n + R_{\text{oracle}}} \sim 2 \log d.
\]

If \( \beta \) is sparse, so 0 except for \( r \ll d \) large components, Then, \( R_{\text{oracle}} = r/n \). So Lasso has small risk in sparse cases.
Nonparametric Regression

Given \((X_1, Y_1), \ldots, (X_n, Y_n)\) where

\[
Y_i \in \mathbb{R}, \quad X_i = (X_{1i}, \ldots, X_{di})^T \in \mathbb{R}^d,
\]

\[
Y_i = m(X_{1i}, \ldots, X_{di}) + \epsilon_i, \quad \mathbb{E}(\epsilon_i) = 0
\]

Risk:

\[
R(m, \hat{m}) = \int \mathbb{E}(\hat{m}(x) - m(x))^2 dx
\]

Minimax theorem:

\[
\inf_{\hat{m}} \sup_{m \in \mathcal{F}} R(m, \hat{m}) \asymp \left( \frac{1}{n} \right)^{4/(4+d)}
\]

where \(\mathcal{F}\) is class of functions with 2 smooth derivatives. Note the curse of dimensionality.
Plotting the Curse

$d = 20$

Risk

Risk = 0.01

$d = 20$
Simplest Nonparametric Estimator

Kernel regression (Nadaraya-Watson)

\[ \hat{m}_h(x) = \frac{\sum_{i=1}^{n} K_h(x_i - x) Y_i}{\sum_{i=1}^{n} K_h(x_i - x)} = S_x Y \]

\[ K_h(x, x') \propto \exp \left( - \sum_{j=1}^{d} \frac{(x_j - x_j')^2}{2h_j^2} \right) \]
Local Linear Regression

For \( u \) near \( x \):

\[
m(u) \approx a_0(x) + a_1(x)(u - x)
\]

Define:

\[
\hat{m}(x) = \left( \hat{a}_0(x) + \hat{a}_1(x)(u - x) \right)_{u=x} = \hat{a}_0(x)
\]

where \( \hat{a} \) minimizes the local sums of squares:

\[
\sum_{i=1}^{n} (Y_i - a_0(x) - a_1(x)(X_i - x))^2 \left( K \left( \frac{x - X_i}{h} \right) \right)
\]

Example:

\[
K(x) = e^{-x^2/2}.
\]

Bandwidth \( h \) controls amount of smoothing. The estimator is insensitive to the choice of \( K \) but is highly sensitive to the choice of \( h \).
Sparse Regression

- In many applications, reasonable to expect true function depends only on small number of variables
- Assume
  \[ m(x) = m(x_R) \]
  where \( x_R = (x_j)_{j \in R} \) are the relevant variables with \( |R| = r \ll d \)
- Can hope to achieve the better minimax rate \( n^{-4/(4+r)} \)
- Challenge: Variable selection in nonparametric regression
Rodeo: The Main Idea

- Use a nonparametric estimator based on a kernel
- Start with large bandwidths in each dimension, for an estimate having small variance but high bias
  - Choosing large bandwidth is like ignoring a variable
- Compute the derivatives of the estimate with respect to bandwidth
- Threshold the derivatives to get a sparse estimate

Intuition: If a variable is irrelevant, then changing the bandwidth in that dimension should only result in a small change in the estimator
Rodeo: The Main Idea

![Diagram showing Rodeo path, Ideal path, and Optimal bandwidth]
Rodeo: Regularization of derivative expectation operator

\[ \tilde{m}(x) = \hat{m}_1(x) - \int_0^1 \langle \hat{D}(h(s)), \dot{h}(s) \rangle ds \]

\[ D(h) = \nabla \mathbb{E}(\hat{m}_h(x)) \]
Using Local Linear Regression

The estimator can be written as

$$
\hat{m}_h(x) = \sum_{i=1}^{n} G(X_i, x, h)Y_i
$$

Our method is based on the statistic

$$
Z_j = \frac{\partial \hat{m}_h(x)}{\partial h_j} = \sum_{i=1}^{n} G_j(X_i, x, h)Y_i
$$

The estimated variance is

$$
s_j^2 = \text{Var}(Z_j | X_1, \ldots, X_n) = \sigma^2 \sum_{i=1}^{n} G_j^2(X_i, x, h)
$$
Rodeo: Hard Tresholding Version

1. Select parameter $0 < \beta < 1$ and initial bandwidth $h_0$.

2. Initialize the bandwidths, and activate all covariates:
   (a) $h_j = h_0$, $j = 1, 2, \ldots, d$.
   (b) $A = \{1, 2, \ldots, d\}$

3. While $A$ is nonempty, do for each $j \in A$:
   (a) Compute estimated derivative expectation: $Z_j$ and $s_j$
   (b) Compute threshold $\lambda_j = s_j \sqrt{2 \log(nc_n)}$.
   (c) If $|Z_j| > \lambda_j$, set $h_j \leftarrow \beta h_j$; otherwise remove $j$ from $A$.

4. Output bandwidths $h^* = (h_1, \ldots, h_d)$ and estimator
   $$\tilde{m}(x) = \hat{m}_{h^*}(x)$$
Example: $m(x) = 5x_1^2x_2^2$, $d = 10$
Example: \( m(x) = 2(x_1 + 1)^3 + 2 \sin(10x_2), \ d = 20 \)
Loss with $r=2$, Increasing Dimension

Leave-one-out cross-validation

Rodeo
One Dimensional Example
Our goal is to show that:

- Bandwidths of the relevant variables shrink (but not too much)
- Bandwidths of the irrelevant variables stay (relatively) large

Keep in mind: we’re making a new fit at each test point (so we actually need only assume local sparsity.)

- We need to make assumptions about the function and the sampling density
Theorem. Assume that $d \log d = O(\log n)$, $r = O(1)$, and $h_0 = \frac{c_0}{\log \log n}$ for some $c_0 > 0$. Define

$$L_j^{(s)} = \begin{cases} \frac{\nu_2 m_{j,j}(x)}{f(x)} h_j^{(s)} & j \leq r \\ - \text{tr} \left( H_R^{(s)} H_R^{(s)} \right) \nu_2^2 (\nabla_j \log f(x))^2 h_j^{(s)} & j > r. \end{cases}$$

Then, for $T_n \leq c_1 \log n$,

$$\mathbb{P} \left( \max_{1 \leq j \leq d, 1 \leq s \leq T_n} |\mu_j^{(s)} - L_j^{(s)}| > \epsilon \right) \to 0,$$

for all $\epsilon > 0$. 

Analysis
Analysis

**Theorem.** Suppose that $d \log d = O(\log n)$ and $m_{jj}(x) \neq 0$ for all $j \leq r$. Then the rodeo outputs bandwidths $h^*$ that satisfy

$$\mathbb{P}\left(h_j^* = h_0 \text{ for all } j > r\right) \to 1$$

and the risk of the estimator satisfies

$$\mathcal{R}(h^*) = O_P\left(n^{-\frac{4}{4+r} + \epsilon}\right)$$

for every $\epsilon > 0$. 
Recall: Lasso/Basis Pursuit
(Chen & Donoho, 1994; Tibshirani, 1996)

\[ \sum_{j=1}^{d} |\beta_j| \leq t \] Level sets of squared error

For orthogonal designs, solution given by soft thresholding

\[ \hat{\beta}_j = \text{sign}(\beta_j) \left( |\beta_j| - \lambda \right)_+ \]
Soft Thresholding

Using a lasso in the rodeo:

Replace \( \hat{D} \) with the soft-thresholded estimate

\[
\hat{D}_j(t) = \text{sign}(Z_j)(|Z_j| - \lambda_j)_+
\]

Estimator is then

\[
\tilde{m}(x) = \hat{m}_{h_0}(x) - \int_0^1 \langle D(s), \dot{h}(s) \rangle ds
\]

\[
\approx \hat{m}_{h_0}(x) - \sum_{s=1}^{t} \langle \hat{D}(s), dh(s) \rangle
\]
Hard vs. Soft Thresholding

![Graph showing the comparison between hard and soft thresholding]

The graph illustrates the difference in loss between hard and soft thresholding across different frequency values. The frequency distribution is shown on the left, with a bar chart indicating the relative frequency of loss values. On the right, a box plot provides a visual summary of the distribution, highlighting the central tendency and spread of the data.
Greedy Rodeo and LARS

• Rodeo can be viewed as a nonparametric version of least angle regression (LARS), (Efron et al., 2004)

• In forward stagewise, variable selection is incremental. LARS adds the variable most correlated with the residuals of the current fit.

• The Lasso can be obtained as a simple modification of LARS

• For the Rodeo, the derivative is essentially the correlation between the output and the derivative of the effective kernel

• Reducing the bandwidth is like adding more of that variable
Greedy Rodeo on Diabetes Data

Rodeo order: 3 (body mass index), 9 (serum), 7 (serum), 4 (blood pressure), 1 (age), 2 (sex), 8 (serum), 5 (serum), 10 (serum), 6 (serum).

LARS order: 3, 9, 4, 7, 2, 10, 5, 8, 6, 1.
Extensions

• Density estimation
• Local polynomial estimation
• Classification using rodeo with generalized linear models
• Other nonparametric estimators
• Time series models, graphical models
Sparse Density Estimation

(Recent work with Han Liu and Larry Wasserman)

Our sparsity assumption:

\[ f(x) \propto g(x_R) h(x) \quad \text{where } h_{jj}(x) = o(1) \text{ for } j \not\in R \]

Kernel density estimator:

\[
\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} \frac{1}{h_j} K \left( \frac{x_j - X_{ij}}{h_j} \right)
\]
Density Estimation

Using KDE2d

Using Rodeo
Density Estimation
Reverse Rodeo for Image Densities

Test point

Bandwidths during reverse rodeo
(Dark color, small bandwidth)
Density Estimation
Briefly: Learning Sparse Graphical Models from Data

- Recent work with Pradeep Ravikumar (CMU) and Martin Wainwright (Berkeley)
- Problem: Learn the graph of a discrete Markov random field from examples
- On the heels of recent flurry of work in $\ell_1$-regularization for sparsity
Graph Learning

\[
p(x | \theta) \propto \exp \left( \sum_{s \in V} \theta_s x_s + \sum_{s,t \in E} \theta_{s,t} x_s x_t \right), \quad x_v \in \{0, 1\}
\]

- Suppose we observe samples from a graphical model, but the graph is unknown
- Can we learn the graph from the data?
- In general, the problem is NP-hard
- Gaussian case recently tackled using the Lasso (Meinshausen and Bühlman, 2006)
Graph Learning: Discrete Case

Given \( n \) samples \( x^{(i)} \in \{0, 1\}^p \) drawn from an unknown distribution \( p(x; \theta^*) \), of the form

\[
p(x; \theta) = \exp \left( \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{s,t} x_s x_t - \Psi(\theta) \right)
\]

the goal is determine the set of edges in the graph, equivalently, the neighbors \( \mathcal{N}(s) \).
Graph Learning

Optimization problem

\[ \hat{\theta}_s, \lambda = \arg \min_{\theta \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left[ \log(1 + \exp(\theta^T z^{(i,s)})) - x_s^{(i)} \theta^T z^{(i,s)} \right] + \lambda n \| \theta \|_1 \right\}. \]

where \( s \in V \), and \( z^{(i,s)} \in \{0, 1\}^p \) denotes the vector where \( z_t^{(i,s)} = x_t^{(i)} \) for \( t \neq s \) and \( z_s^{(i,s)} = 1 \).

Our estimate of the neighborhood \( \mathcal{N}(s) \) is then given by

\[ \hat{\mathcal{N}}_n(s) = \left\{ t \in V, t \neq s : \hat{\theta}_t^{s,\lambda} \neq 0 \right\}. \]
Graph Learning

**Theorem.** Suppose that the regularization parameter $\lambda_n$ is chosen such that (a) $n\lambda_n^2 - 2\log(p) \to +\infty$, and (b) $d_{\text{max}}\lambda_n \to 0$. Then

$$
P \left( \hat{N}_n(s) = \mathcal{N}(s), \forall s \in V_n \right) \to 1
$$

- Number of nodes can grow as $p = O(n^\gamma)$ for any $\gamma$.
- We require an “incoherence” condition on the Fisher information matrix: $\|Q_{ScS}Q_{SS}^{-1}\|_\infty < 1 - \epsilon$. 
Summary

• Sparsity is playing an increasingly important role in statistics and machine learning as data increases in complexity and dimension.
  - In order to be “learnable,” there must be lower-dimensional structure. Challenge is to detect and extract this structure.

• Rodeo is conceptually simple and practical, and has theoretically nice properties.

• $\ell_1$-regularization for learning discrete graphical models.