Fast Sparse Regression and Classification

Jerome H. Friedman

Stanford University
PREDICTION (Regression/Classification)

\[ y = \text{outcome/response variable} \]

\[ \mathbf{x} = \{x_1, \ldots, x_n\} \text{ predictors} \]

Goal: \( \hat{y} = F(\mathbf{x}) \)

Want good \( F(\mathbf{x}) \)
LINEAR MODEL

\[ F(x; \mathbf{a}) = a_0 + \sum_{j=1}^{n} a_j x_j \]

\( a_0 = \text{intercept} \)

\( \{a_j\}_1^n = \text{coefficients} \)
ACCURACY

Cost for error: \( L(y, F) \)

\[
L(y, F) = (y - F)^2 : y \in R
\]

\[
L(y, F) = \log(1 + e^{-yF}) : y \in \{-1, 1\}
\]

\[
L(y, F) = \log \left( \sum_{k=1}^{K} e^{F_k} \right) - \sum_{k=1}^{K} I(y = c_k) F_k
\]

\[ y \in \{c_1, c_2, \ldots, c_K\} \]

many many more
PREDICTION RISK

\[ R(a) = E_{x,y}L(y, F(x; a)) \]

Optimal solution: \( a^* = \arg \min_a R(a) \)

\( p(x, y) \) unknown \( \Rightarrow a^* \) unknown
STATISTICAL (MACHINE) LEARNING

Training data: \( \{y_i, x_i\}_{1}^{N} \sim p(x, y) \)

\[
\hat{R}(\mathbf{a}) = \frac{1}{N} \sum_{i=1}^{N} L \left( y_i, a_0 + \sum_{j=1}^{n} a_j x_{ij} \right)
\]

\( \hat{\mathbf{a}} = \text{arg min}_{\mathbf{a}} \hat{R}(\mathbf{a}) \)

If not \( N >> n \), not very good!

\[
R(\hat{\mathbf{a}}) >> R(\mathbf{a}^*) \quad \text{(high variance)}
\]
REGULARIZATION (biased learning)

\[ \hat{a}(t) = \arg\min_a \hat{R}(a) \text{ s.t. } P(a) \leq t \]

\( P(a) \geq 0 \) constraining function

\( t \geq P(\hat{a}) \): no constraint \( \Rightarrow \) no bias / max. variance

\( t = 0 \): max. constraint \( \Rightarrow \) max. bias / min. variance

\( 0 < t < P(\hat{a}) \Rightarrow \) bias–variance trade–off
EQUIVALENT FORMULATION

\[ \hat{a}(\lambda) = \arg \min_a [\hat{R}(a) + \lambda \cdot P(a)] \]

Here \( P(a) = “\text{penalty}“ \)

\[ \infty \leq \lambda \leq 0 \sim 0 \leq t \leq P(\hat{a}) \]

\( \hat{a}(\lambda) \sim 1\text{-dim. path of solutions} \in S^{n+1} \)

\( S^{n+1} = \text{parameter space} \)
MODEL SELECTION ($\lambda$)

$$\lambda^* = \arg \min_{0 \leq \lambda \leq \infty} R(\hat{a}(\lambda))$$

Model selection criterion:

$$\tilde{R}(a) = \text{surrogate for } R(a)$$

depends on $L(y, F')$ & $P(a)$

$$\hat{\lambda} = \arg \min_{0 \leq \lambda \leq \infty} \tilde{R}(\hat{a}(\lambda))$$

$$\hat{a}(\hat{\lambda}) = \text{selected model}$$

Cross-validation: any $L(y, F')$ & $P(a)$
PENALTY SELECTION

\[ a^* = \text{point in } S^{n+1} \]

Choose penalty that induces paths that

on average come close to \( a^* \)

\[ \{y_i, x_i\}_1^N \sim p(x, y) \]

Depends on \( a^* \)

Choose \( P(a) \) based on knowledge of \( a^* \)
SPARSITY

Fraction of non influential variables

\[ S(a) = \frac{\#(|a_k| = \text{small})}{n} \]

Assumption: \( \hat{a} \sim a^* \Rightarrow S(\hat{a}) \sim S(a^*) \)
Choose $P(a)$ s.t. $S(\hat{\lambda}(\lambda^*)) \simeq S(a^*)$

Don’t know $S(a^*)$?

Family of penalties $P_\gamma(a)$: $\gamma$ regulates $S(\hat{a})$

bridging sparse $\rightarrow$ dense solutions

Model selection to jointly estimate $(\gamma, \lambda)$

(“bridge regression”: Frank & Friedman 1993)
POWER FAMILY

\[ P_\gamma(a) = \sum_{j=1}^n |a_j|^{\gamma} \]

With \( L(y, F) = (y - F)^2 \):

\[ \gamma = 2 : \text{ ridge–regression (dense)} \]

\[ \gamma = 1 : \text{ lasso (moderately sparse)} \]

\[ \gamma = 0 : \text{ (all) subsets selection (sparsest)} \]

0 \leq \gamma \leq 2 \text{ bridges subset } \rightarrow \text{ ridge}

Note: \( \gamma \geq 1 \Rightarrow \text{ convex}, \ \gamma < 1 \Rightarrow \text{ non convex} \)
Generalized Elastic Net

$1 \leq \beta \leq 2$ (convex: lasso $\rightarrow$ ridge):

Elastic Net (Zou & Hastie 2005)

$$P_\beta(a) = \sum_{j=1}^{n} \left( \beta - 1 \right) \frac{a_j^2}{2} + (2 - \beta) |a_j|$$

$0 \leq \beta < 1$ (non convex: subset selection $\rightarrow$ lasso):

$$P_\beta(a) = \sum_{j=1}^{n} \log((1 - \beta) |a_j| + \beta)$$

$0 \leq \beta \leq 2$ bridges subset $\rightarrow$ ridge

Better statistical & computational properties

Method works for both + many more
Power family

Generalized elastic net
BRIDGE REGRESSION

(1) Repeatedly solve:

$$\hat{a}_\beta(\lambda) = \arg \min_a [\hat{R}(a) + \lambda \cdot P_\beta(a)]$$

$$0 \leq \beta \leq 2, \quad 0 \leq \lambda \leq \infty$$

(2) $$(\hat{\beta}, \hat{\lambda}) \leftarrow \text{model selection criterion}$$

(3) $$\hat{a}_{\hat{\beta}}(\hat{\lambda}) = \text{solution}$$

Big challenge: fast enough algorithm for (1)

Especially for $P_\beta(a) = \text{non convex}$
DIRECT PATH SEEKING

Goal: rapidly produce path $\sim$ given $P(a)$

without repeatedly optimizing

$\nu \geq 0$: path length; $\Delta \nu > 0$: increment

$d(\nu) =$ direction in parameter space
Algorithm

Initialize: $\nu = 0; \quad \hat{a}(0) = 0$

Loop {

$\hat{a}(\nu + \Delta \nu) = \hat{a}(\nu) + \mathbf{d}(\nu) \cdot \Delta \nu$

$\nu \leftarrow \nu + \Delta \nu$

}

Until $(\hat{R}(\hat{a}(\nu)) = \text{min})$

Methods differ: $\mathbf{d}(\nu) \& \Delta \nu$
EXAMPLES

\[ L(y, F) = (y - F)^2: \]

PLS \sim \text{ridge–regression} (\beta = 2)

LAR \sim \text{lasso} (\beta = 1)

Forward stepwise \sim \text{all–subsets} (\beta = 0)

Any convex \( L(y, F) : \)

Gradient boosting \sim \text{lasso} (\beta = 1)

Want bridge regression: \( 0 \leq \beta \leq 2 \)
Generalized Path Seeking (GPS)

Fast algorithm for:

(1) any convex $L(y, F)$ (some non convex)

(2) any $P(a)$ s.t. $\frac{\partial P(a)}{\partial |a_j|} \geq 0$

i.e. $P(a)$ monotone $\uparrow |a_j|$
EXAMPLES

power family

generalized elastic net family (*)

SCAD, MC+ family

grouped lasso, grouped bridge family, CAP

many more

extend to bigger problems

any convex loss
Definitions

$\nu \geq 0$: path length

$\Delta \nu > 0$: small increment

$g_j(\nu) = - \left[ \frac{\partial \hat{R}(a)}{\partial a_j} \right]_{a=\hat{a}(\nu)}$

$p_j(\nu) = \left[ \frac{\partial P(a)}{\partial |a_j|} \right]_{a=\hat{a}(\nu)}$

$\lambda_j(\nu) = g_j(\nu) / p_j(\nu)$
1 Initialize: \( \nu = 0; \{ \hat{a}_j(0) = 0 \}^n_1 \)

2 Loop \{ 

3 Compute \( \{ \lambda_j(\nu) \}^n_1 \)

4 \( S = \{ j \mid \lambda_j(\nu) \cdot \hat{a}_j(\nu) < 0 \} \)

5 if \( S = \) empty \( j^* = \arg \max_j |\lambda_j(\nu)| \)

6 else \( j^* = \arg \max_{j \in S} |\lambda_j(\nu)| \)

7 \( \hat{a}_{j^*}(\nu + \Delta \nu) = \hat{a}_{j^*}(\nu) + \Delta \nu \cdot \text{sign}(\lambda_{j^*}(\nu)) \)

8 \( \{ \hat{a}_j(\nu + \Delta \nu) = \hat{a}_j(\nu) \}_{j \neq j^*} \)

9 \( \nu \leftarrow \nu + \Delta \nu \)

10 } Until \( \lambda(\nu) = 0 \)
THEOREM

\( \hat{a}(\lambda) = \text{exact path} \)

\( \hat{a}(\nu) = \text{GPS path} \)

If for all \( \lambda > \lambda_0 \)

all \( \{\hat{a}_j(\lambda)\}_{1}^{n} \) are continuous and monotone

Then for all \( \lambda > \lambda_0 \)

\( \hat{a}(\nu) = \hat{a}(\lambda), \text{ as } \Delta \nu \rightarrow 0 \)

i.e. GPS produces exact path
Otherwise: \( \hat{a}_j(\nu) \simeq \hat{a}_j(\lambda) \)

When \( \hat{a}_j(\lambda) \) becomes non monotone:

\( \hat{a}_j(\nu) \) tends to slightly delay becoming non monotone

When \( \hat{a}_j(\lambda) \) discontinuous (\( \gamma < 1, \beta < 1/2 \)):

\( \hat{a}_j(\nu) = \) continuous (by construction)

\( \sim \) interpolates between \( \hat{a}_j(\lambda) \) discontinuities
Regression: \[ L(y, F) = (y - F)^2 \]

Diabetes data: \( n = 10, \ N = 442 \)

Used in LARS (Efron et al 2002)

red = exact (convex), black = GPS
Logistic Regression / Classification

\[ L(y, F) = \log(1 + e^{-yF}) \]

South African heart transplant data

\[ n = 9, \quad N = 462 \]

\[ y \in \{1, -1\} = \{\text{success, failure}\} \]

red = exact (convex), black = GPS
Explained deviance Coefficients

Elastic Net 1.9

Elastic Net 1.5

Elastic Net 1.25

Lasso
Explained deviance

Coefficients

Lasso

Elastic Net 0.5

Elastic Net 0.25

Elastic Net 0.0
Regression: under-determined example

\[ n = 10000, \quad N = 200 \]

\[ x_i \sim N(0, C); \quad C_{jj} = 1, \quad C_{jk} = 0.4 \]

\[ y_i = \sum_{j=1}^{n} a_j^* x_{ij} + \varepsilon_i \]

\[ \varepsilon_i \sim N(0, \sigma^2); \quad \sigma \sim 3/1 \text{ signal/noise} \]

\[ |a_j^*| = [31 - j]_+, \quad \text{sign}(a_j^* + 1) = -\text{sign}(a_j^*) \]
\( \beta \in \{1.9, 1.7, 1.5, 1.0 \text{ (lasso, blue)}, 0.5, 0.3, 0.2, 0.1, \ldots \} \)
Logistic Regression / Classification

under-determined example

\[ n = 10000, \quad N = 200 \]

\[ \mathbf{x}_i \sim N(0, \mathbf{C}); \quad C_{jj} = 1, \quad C_{jk} = 0.4 \]

\[ y_i \in \{0, 1\}; \quad \text{log-odds} = \sum_{j=1}^{n} a_j^* x_{ij} \]

\[ |a_j^*| = \rho \cdot [16 - j]_+, \quad \text{sign}(a_{j+1}^*) = -\text{sign}(a_j^*) \]

\[ \rho \sim 5\% \text{ error rate} \]
\[ \beta \in \{1.9, 1.7, 1.5, 1.0 \text{ (lasso, blue)}, 0.7, 0.5, 0.3, 0.0\} \]
THEREFORE

\[ P_\beta(a) = \text{generalized elastic net} \]

\[ \beta \downarrow \Rightarrow S(\hat{a}) \uparrow \text{monotonically} \]

at all path points
Penalty Selection ($\beta$)

Regression: under-determined example

$$n = 10000, \quad N = 200$$

50 data sets $\sim p(x, y)$

Distribution of closest “distance” to truth $a^*$ (risk)

Methods:

GPS: $\beta \in \{0.0, 0.1, 0.2, 0.5\}$

forward stepwise, lasso, elastic net (1.5)
Corr = 0.4

Corr = 0.0

Corr = 0.4, a > 0
Corr = 0.4

Corr = 0.0

Corr = 0.4, \( a > 0 \)
Post-processing Selectors

(1) \( \hat{a}(\lambda) = \arg \min_a \hat{R}(a) + \lambda P(a) \)

\( P(a) = \text{convex (lasso)} \)

(2) \( A(\lambda) = \{j\} \hat{a}_j(\lambda) \neq 0 \) (active variables at \( \lambda \))

(3) \( \hat{a}(\lambda) = \arg \min_a \hat{R}(a) \) s.t. \( \{a_j = 0\}_{j \notin A(\lambda)} \)

Intuition (sparse problems):

\( \hat{a}(\lambda) \simeq \text{selects correct variables} \)

but over shrinks their values
CONCLUSIONS

(1) best penalty (prior) for \(|a_j|\) depends on
\(|a^*_j|, \text{sign}(a^*_j)| \text{ and } x - \text{distribution}

(2) need bridge regression to chose \((\hat{\beta}, \hat{\lambda})\)

(3) when sparse non convex \(P(a)\) is best:
   better variable selection & shrinkage

(4) best direct methods \(\rightarrow\) best selectors

(5) results same for logistic regression
Generalized Path Seeking

For same $L(y, F)$ & $P(a)$:

paths close to exact solutions

same sparseness properties

Can be applied with:

non convex $P(a) \Rightarrow$ sparser than lasso

any convex $L(y, F')$, some non convex

Used as selector $\rightarrow$ further improvement

Multinomial regression
Speed

\[ n = 10000, \; N = 200 \]

Solutions at 500 path points \( \simeq 0.5 \text{ sec.} \) (non/convex)

Bridge regression:

\[ 6 \beta \text{- values} \times 10\text{-fold xval}: \sim 30 \text{ sec} \]

equivalent to solving 30000 optimization problems

(most non convex)

Computation scales \( \sim \) linearly with \( n \& N \) (\( n \gg N \))
TALK


PAPER