Simple Variable Selection LASSO: Sparse Regression

Machine Learning – CSE546

Carlos Guestrin
University of Washington

October 7, 2013

Sparsity

$|w| = 100B$

- Vector **w** is sparse, if many entries are zero:

  $\hookrightarrow (1.7, -2.2, 0, 0, 0, 0, 3.3, 0, 0, 0 \ldots)$

- Very useful for many tasks, e.g.,
  - **Efficiency**: If size(**w**) = 100B, each prediction is expensive: $\hat{f}(x) = w_0 + \sum_i w_i h_i(x)$ — 100B computations
    - If part of an online system, too slow
    - If **w** is sparse, prediction computation only depends on number of non-zeros
  - **Interpretability**: What are the relevant dimension to make a prediction?
    - E.g., what are the parts of the brain associated with particular words?

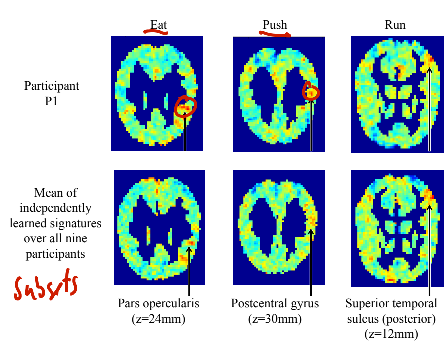


Figure from Tom Mitchell

with k non-zero

- But computationally intractable to perform "all subsets" regression $\binom{100B}{k}$ subsets

Simple greedy model selection algorithm

- Pick a dictionary of features
  - e.g., polynomials for linear regression
- Greedy heuristic:
  - Start from empty (or simple) set of features $F_0 = \varnothing$ ← or constant
  - Run learning algorithm for current set of features F_t
    - Obtain h_t coeff w_t
  - Select **next best feature $\mathbf{X_i^*}$**
    - e.g., X_j that results in lowest training error learner when learning with $F_t + \{X_j\}$
  - $F_{t+1} \leftarrow F_t + \{X_i^*\}$
  - Recurse

Greedy model selection

- Applicable in many settings:
  - Linear regression: Selecting basis functions
  - Naïve Bayes: Selecting (independent) features $P(X_i|Y)$
  - Logistic regression: Selecting features (basis functions)
  - Decision trees: Selecting leaves to expand
- Only a heuristic!
  - But, sometimes you can prove something cool about it
    - e.g., [Krause & Guestrin '05]: Near-optimal in some settings that include Naïve Bayes
- There are many more elaborate methods out there

When do we stop???

- Greedy heuristic:
  - …
  - Select **next best feature X_i^***
    - e.g., X_j that results in lowest training error learner when learning with $F_t + \{X_j\}$
  - $F_{t+1} \leftarrow F_t + \{X_i^*\}$
  - Recurse

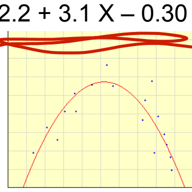
When do you stop???

- ~~When training error is low enough?~~
- ~~When test set error is low enough?~~
- Cross validation, please!

Regularization in Linear Regression

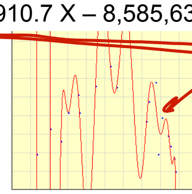
- Overfitting usually leads to very large parameter choices, e.g.:

$-2.2 + 3.1\ X - 0.30\ X^2$



$-1.1 + 4{,}700{,}910.7\ X - 8{,}585{,}638.4\ X^2 + \dots$

penalty for large weights



overfitting

- ***Regularized*** or ***penalized*** regression aims to impose a "complexity" penalty by penalizing large weights
  - "Shrinkage" method

L_2 regularization → penalizes towards smoother functions

Variable Selection by Regularization

$\sum_i w_i^2$

- Ridge regression: Penalizes large weights
- What if we want to perform "feature selection"?
  - E.g., Which regions of the brain are important for word prediction?
  - Can't simply choose features with largest coefficients in ridge solution

lots of small coeffs, rather than a few large ones

- Try new penalty: Penalize non-zero weights
  - Regularization penalty: $\|W\|_1 = \sum_i |w_i|$

LASSO

  - Leads to sparse solutions
  - Just like ridge regression, solution is indexed by a continuous param λ
  - This simple approach has changed statistics, machine learning & electrical engineering

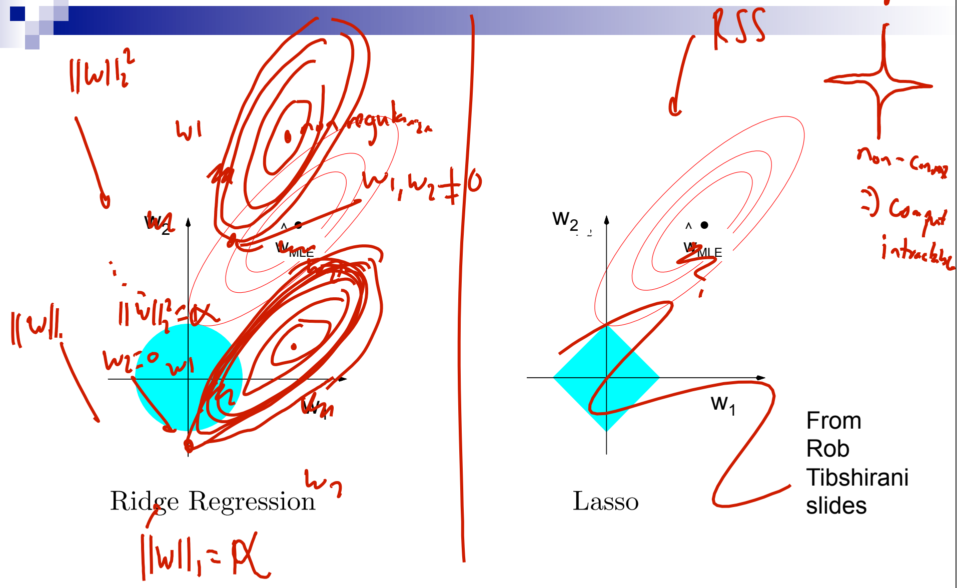
LASSO Regression

- **LASSO:** least absolute shrinkage and selection operator
- New objective:

$$\min_w \underbrace{\sum_{j=1}^{N}\left(t(x_j) - \left(w_0 + \sum_{i=1}^{k} w_i h_i(x_j)\right)\right)^2}_{RSS} + \lambda \sum_{i=1}^{k} |w_i|$$

please don't penalize the poor w_0, it did nothing to you

Geometric Intuition for Sparsity



From Rob Tibshirani slides

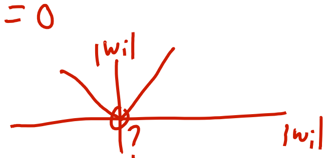
Optimizing the LASSO Objective

- LASSO solution:

$$\hat{\mathbf{w}}_{LASSO} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

Take derivative & set = 0

1. Derivative of $|w_i|$



2. Even if you could take derivative, no closed-form solution to $\hat{w}_{LASSO}$

Coordinate Descent

- Given a function F
  - Want to find minimum

    $\hat{w} = \arg\min F(w_0, w_1, \ldots w_k)$
- Often, hard to find minimum for all coordinates, but easy for one coordinate

  1.d optimization problem — fixing others
- Coordinate descent: initialize $w = 0$ or something else

  while not converged

  pick coordinate ℓ — fix values from previous iteration

  $\hat{w}_\ell \leftarrow \arg\min_{\omega} F(w_0, w_1, \ldots w_{\ell-1}, \omega, w_{\ell+1}, \ldots w_k)$

  Converges !! :)

  - but in many problems, to local optimal only

  · but LASSO & other "separable" convex problems, global optimum
- How do we pick next coordinate?

  random, round robin, "smartly"
- Super useful approach for *many* problems
  - Converges to optimum in some cases, such as LASSO

Optimizing LASSO Objective One Coordinate at a Time

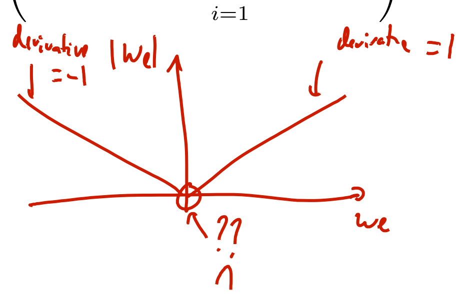
$$\overbrace{\sum_{j=1}^{N}\left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j))\right)^2}^{RSS(w)} + \overbrace{\lambda \sum_{i=1}^{k} |w_i|}^{\text{log}}$$

- Taking the derivative:
  - Residual sum of squares (RSS):

$$\frac{\partial}{\partial w_\ell} RSS(\mathbf{w}) = -2 \sum_{j=1}^{N} h_\ell(x_j) \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j))\right)$$

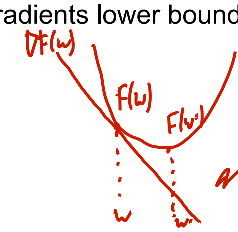
  - Penalty term:

$$\frac{\partial}{\partial w_\ell} \lambda \sum_{i=1}^{k} |w_i| = \lambda \frac{\partial}{\partial w_\ell} |w_\ell|$$



Subgradients of Convex Functions

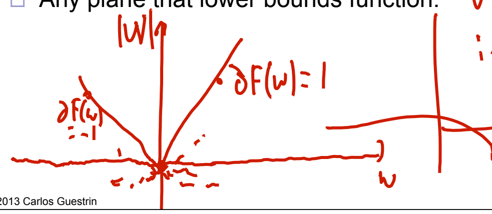
- Gradients lower bound convex functions:



$F(w') \geq F(w) + \nabla F(w)^T (w'-w)$

- Gradients are unique at **w** iff function differentiable at **w**

- Subgradients: Generalize gradients to non-differentiable points:
  - Any plane that lower bounds function: v is subgradient F at w if $F(w') \geq F(w) + v^T(w'-w)$ $\forall w'$



at 0: $\partial_{w_\ell} F(0) = [-1, 1]$

Set to 0: optimum when $0 \in \partial_w F(w)$

Taking the Subgradient

$$\partial_{w_\ell}\Big[\underbrace{\sum_{j=1}^{N}\left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j))\right)^2}_{RSS} + \underbrace{\lambda \sum_{i=1}^{k} |w_i|}_{\text{Penalty}}\Big]$$

$\frac{\partial}{\partial w_\ell} RSS(w) + \lambda \partial_{w_\ell} |w_\ell| \ni 0$?

- Gradient of RSS term:

$$a_\ell = 2\sum_{j=1}^{N} (h_\ell(\mathbf{x}_j))^2 > 0$$

$$\frac{\partial}{\partial w_\ell} RSS(\mathbf{w}) = a_\ell w_\ell - c_\ell$$

$$c_\ell = 2\sum_{j=1}^{N} h_\ell(\mathbf{x}_j)\left(t(\mathbf{x}_j) - (w_0 + \sum_{i \neq \ell} w_i h_i(\mathbf{x}_j))\right)$$

  - If no penalty ($\lambda = 0$): $\hat{w}_\ell = c_\ell / a_\ell$
- Subgradient of full objective:

$$\partial_{w_\ell} |w_\ell| = \begin{cases} -1 & \text{when } w_\ell < 0 \\ [-1, +1] & \text{when } w_\ell = 0 \\ +1 & \text{when } w_\ell > 0 \end{cases}$$

$$\partial_{w_\ell} F(w) = a_\ell w_\ell - c_\ell + \lambda \partial_{w_\ell} |w_\ell|$$

$$= \begin{cases} a_\ell w_\ell - c_\ell - \lambda & \text{if } w_\ell < 0 \\ [-c_\ell - \lambda, -c_\ell + \lambda] & \text{if } w_\ell = 0 \\ a_\ell w_\ell - c_\ell + \lambda & \text{if } w_\ell > 0 \end{cases}$$

$\Leftarrow$ in optimum, 0 is in this set

Setting Subgradient to 0

$a_\ell > 0$

c_ℓ can be >0, <0, $=0$

$$0 \in \partial_{w_\ell} F(\mathbf{w}) = \begin{cases} a_\ell w_\ell - c_\ell - \lambda & w_\ell < 0 \\ [-c_\ell - \lambda, -c_\ell + \lambda] & w_\ell = 0 \\ a_\ell w_\ell - c_\ell + \lambda & w_\ell > 0 \end{cases}$$

optima with $w_\ell < 0$? $\Rightarrow$ $a_\ell w_\ell - c_\ell - \lambda = 0$ $\Rightarrow$ $w_\ell = \frac{c_\ell + \lambda}{a_\ell} \overset{?}{<} 0$ when $c_\ell < -\lambda$

optima with $w_\ell > 0$? $\Rightarrow$ $a_\ell w_\ell - c_\ell + \lambda = 0$ $\Rightarrow$ $w_\ell = \frac{c_\ell - \lambda}{a_\ell} \overset{?}{>} 0$, when $c_\ell > \lambda$

optima with $w_\ell = 0$ $\Rightarrow$ $0 \in [-c_\ell - \lambda, -c_\ell + \lambda]$ $\Rightarrow$ $-\lambda \le c_\ell \le \lambda$

this why you get sparsity $\Rightarrow$ $w_\ell = 0$

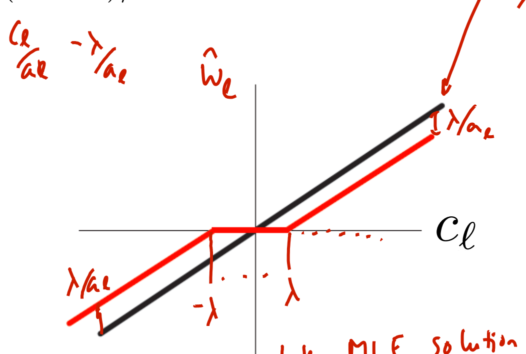
Soft Thresholding

Reminder: MLE for w_ℓ

$\hat{w}_\ell^{MLE} = c_\ell / a_\ell$

$$\hat{w}_\ell = \begin{cases} (c_\ell + \lambda)/a_\ell & c_\ell < -\lambda \\ 0 & c_\ell \in [-\lambda, \lambda] \\ (c_\ell - \lambda)/a_\ell & c_\ell > \lambda \end{cases}$$

c_ℓ / a_ℓ = MLE for $\hat{w}_\ell$



From Kevin Murphy textbook

take MLE solution, threshold to $\emptyset$ when $|c_\ell| \le \lambda$

Coordinate Descent for LASSO (aka Shooting Algorithm)

- Repeat until convergence
  - Pick a coordinate ℓ at (random or sequentially)
    - Set:

$$\hat{w}_\ell = \begin{cases} (c_\ell + \lambda)/a_\ell & c_\ell < -\lambda \\ 0 & c_\ell \in [-\lambda, \lambda] \\ (c_\ell - \lambda)/a_\ell & c_\ell > \lambda \end{cases}$$

minimum of F using subgradient derivation

    - Where:

$$a_\ell = 2\sum_{j=1}^{N}(h_\ell(\mathbf{x}_j))^2$$

$$c_\ell = 2\sum_{j=1}^{N} h_\ell(\mathbf{x}_j)\left(t(\mathbf{x}_j) - (w_0 + \sum_{i\neq\ell} w_i h_i(\mathbf{x}_j))\right)$$

w_0: don't regularize

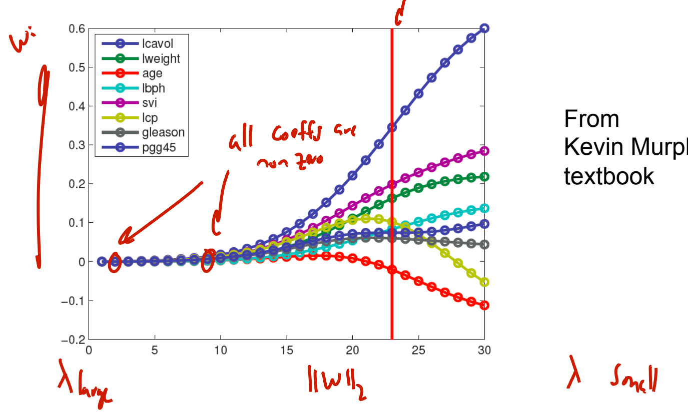
$\hat{w}_0 = c_0/a_0$

$\hat{w}_0 = \frac{1}{N}\sum_{j=1}^{N}\left(t(x_j) - \sum_{i=1}^{k} w_i h_i(x_j)\right)$

  - For convergence rates, see Shalev-Shwartz and Tewari 2009
- Other common technique = LARS
  - Least angle regression and shrinkage, Efron et al. 2004

Recall: *Ridge Coefficient Path*

x val selected λ

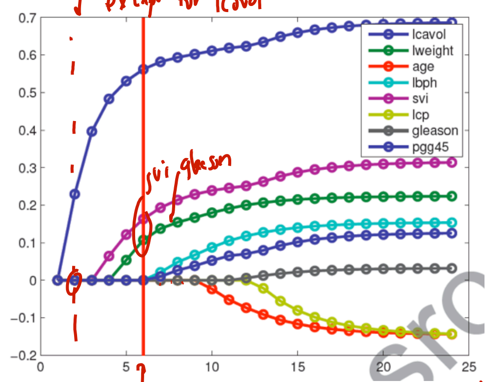


From Kevin Murphy textbook

- Typical approach: select λ using cross validation

Now: *LASSO Coefficient Path*

all coeffs = 0 except for lcavol



λ_{large} λ^* $\|W\|_1$ λ small

From Kevin Murphy textbook

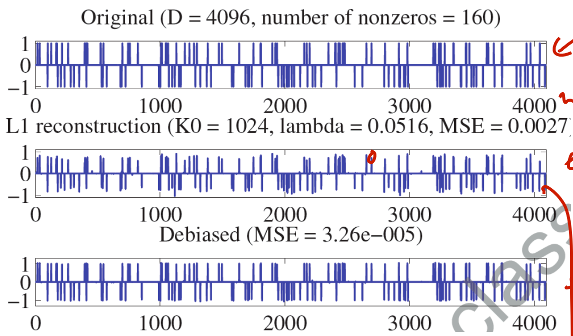
Coeffs or weights can become nonzero than zero again

LASSO Example

Term	Least Squares	Ridge	Lasso
Intercept	2.465	2.452	2.468
lcavol	0.680	0.420	0.533
lweight	0.263	0.238	0.169
age	−0.141	−0.046	
lbph	0.210	0.162	0.002
svi	0.305	0.227	0.094
lcp	−0.288	0.000	
gleason	−0.021	0.040	
pgg45	0.267	0.133	

From Rob Tibshirani slides

Debiasing



true

w

LASSO

many coeff smaller should be

de bias

solve unregularized least squares

to using only selected features

From Kevin Murphy textbook

What you need to know

- Variable Selection: find a sparse solution to learning problem
- L_1 regularization is one way to do variable selection
  - Applies beyond regressions
  - Hundreds of other approaches out there
- LASSO objective non-differentiable, but convex → Use subgradient
- No closed-form solution for minimization → Use coordinate descent
- Shooting algorithm is very simple approach for solving LASSO