Simple Variable Selection LASSO: Sparse Regression

Machine Learning – CSE546

Carlos Guestrin

University of Washington

October 7, 2014

Sparsity

100M – 100Bs of params

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

  ↳ {1.7, 2.2, 0 0, 0 0, 2.9, 0 0 0 0 0 ... , −3.1, ... }

- Very useful for many tasks, e.g.,
  - **Efficiency**: If size(**w**) = 100B, each prediction is expensive:
    - If part of an online system, too slow ← $f(x) = w_0 + \sum_{i=1}^{100B} w_i h_i(x)$ most are zero
    - 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?

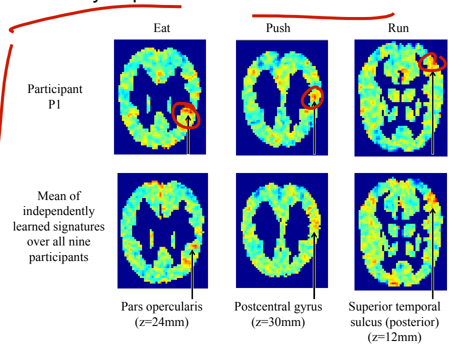


Figure from Tom Mitchell

with K non-zeros

- 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 $F_0 = \emptyset$, or the constant w_0
  - Run learning algorithm for current set of features F_t
    - Obtain h_t
  - Select **next best feature** $\mathbf{X_i^*}$
    - e.g., X_j that results in lowest training error learner when learning with $F_t + \{X_j\}$

100B runs of learning alg. !!

  - $F_{t+1} \leftarrow F_t + \{X_i^*\}$
  - 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

either k features

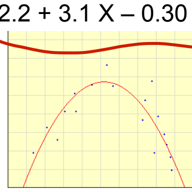
When do you stop???

- ~~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.:

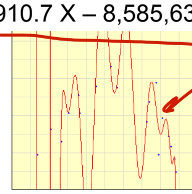
-2.2 + 3.1 X – 0.30 X^2



simple models

-1.1 + 4,700,910.7 X – 8,585,638.4 X^2 + …

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

$\|w\|_2^2 = \sum_i w_i^2$

- Ridge regression: Penalizes large weights with L_2 norm
  → Smooth functions
- 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, but non-zero coeffs → Ridge
    (0.001, 0.1, 3.2, -2.7, -0.02)
- 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:

$$\min_w \overbrace{\sum_{j=1}^N \Big(\underbrace{f(x_j)}_{\text{truth}} - \big(\underbrace{w_0 + \sum_{i=1}^k w_i h_i(x_j)}_{f}\big)\Big)^2}^{RSS} + \lambda \sum_{i=1}^k |w_i|$$

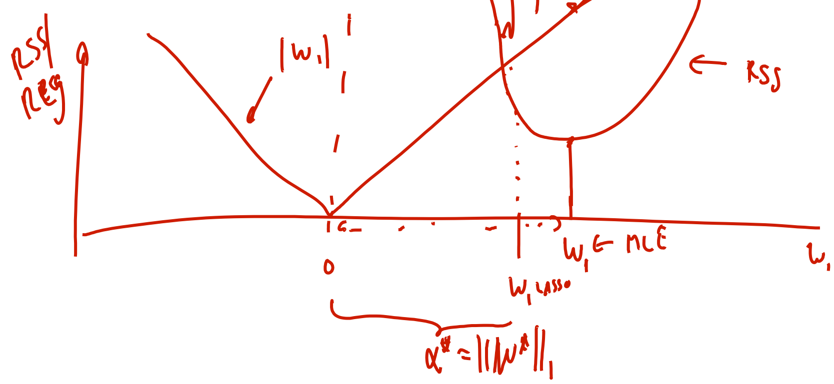
please don't regularize w_0, it didn't do anything to you

Geometric intuition of regularized objectives in 1d

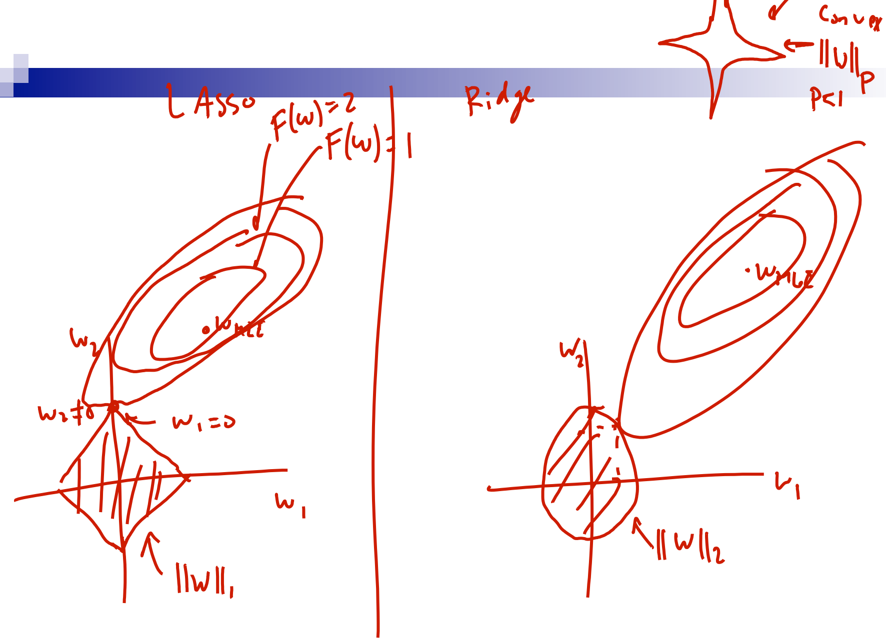
F(w) = RSS + Reg.

- LASSO solution:

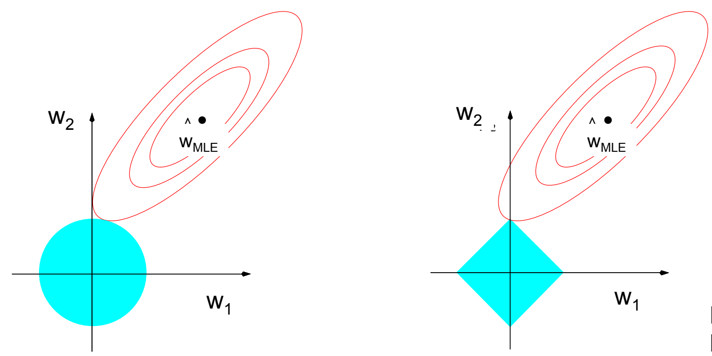
$$\hat{\mathbf{w}}_{LASSO} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$



Geometric intuition



Geometric Intuition for Sparsity



Ridge Regression Lasso

From Rob Tibshirani slides

Optimizing the LASSO Objective

- LASSO solution:

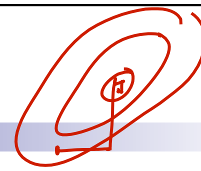
$$\hat{\mathbf{w}}_{LASSO} = \arg\min_{w} \sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

Take the derivative & set to $\emptyset$

1. Derivative of $|w_i|$?

2. Even if you could take derivatives, no closed-form solution exists.

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
- Coordinate descent:

  initialize $w = 0$ or something smarter

  while not converged

  pick a coordinate ℓ

  $\hat{w}_\ell = \arg\min_{w_\ell} F(\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_{\ell-1}, w_\ell, \hat{w}_{\ell+1}, \ldots, \hat{w}_k)$

  fix to values from previous iteration

  converges!! but usually to a local optima
- How do we pick next coordinate?

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

Optimizing LASSO Objective One Coordinate at a Time

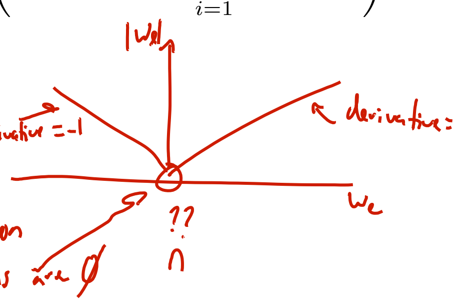
$$\underbrace{\sum_{j=1}^{N} \left(t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2}_{RSS} + \underbrace{\lambda \sum_{i=1}^{k} |w_i|}_{Reg}$$

- 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:

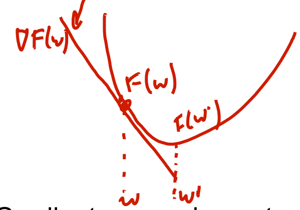
$$\frac{\partial}{\partial w_\ell} \lambda \sum_{i=1}^{k} |w_i| = \lambda \frac{\partial}{\partial w_\ell} |w_\ell|$$



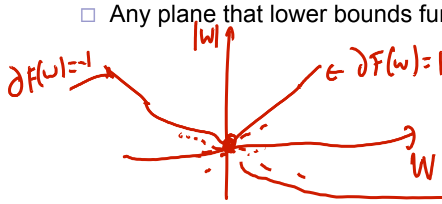
at opt solution most params are 0

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



at 0: $\partial_w F(w) = [-1, 1]$

At optimum, subgradient = 0

Taking the Subgradient

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

- 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 (reg. $\lambda = 0$): $\hat{w}_\ell = c_\ell / a_\ell$
- Subgradient of full objective:

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

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

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

at the OPT w_ℓ this beast is Ø

Setting Subgradient to 0

$a_\ell > 0$

choose w_ℓ such that

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

optimum with $w_\ell < 0$? $\Rightarrow a_\ell w_\ell - c_\ell - \lambda = 0 \Rightarrow 0 > w_\ell = \frac{c_\ell + \lambda}{a_\ell}$ & when $c_\ell + \lambda < 0$ ($c_\ell < -\lambda$)

optimum with $w_\ell > 0$? $\Rightarrow a_\ell w_\ell - c_\ell + \lambda = 0 \Rightarrow 0 < w_\ell = \frac{c_\ell - \lambda}{a_\ell}$ & when $c_\ell - \lambda > 0$ ($c_\ell > \lambda$)

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

$\Rightarrow w_\ell = 0$

↳ sparsity in coord!

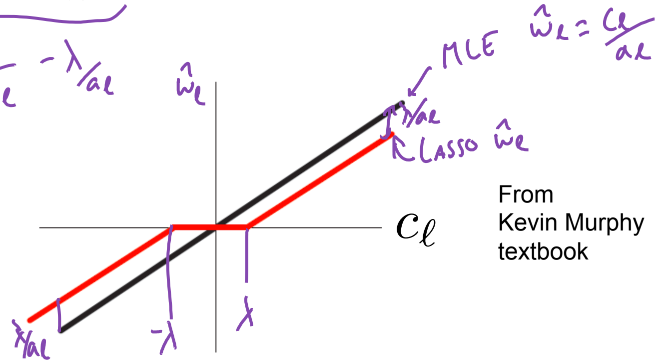
this is why we get sparsity

Soft Thresholding

Reminder, for MLE $\hat{w}_\ell = c_\ell / a_\ell$

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

$\frac{c_\ell}{a_\ell} - \lambda/a_\ell$



From Kevin Murphy textbook

Coordinate Descent for LASSO (aka Shooting Algorithm)

what about w_0?
don't regularize
$\hat{w}_0 = c_0 / a_0$

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

minimum F wrt w_ℓ using subgradient

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

- Where:

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

$$c_\ell = 2 \sum_{j=1}^{N} h_\ell(\mathbf{x}_j) \left(t(\mathbf{x}_j) - (\hat{w}_0 + \sum_{i \neq \ell} \hat{w}_i h_i(\mathbf{x}_j)) \right)$$

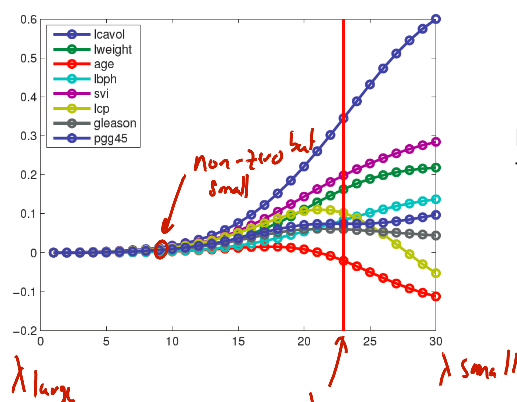
value in previous iteration of all $\hat{w}_i$ except for ℓ

  - 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*

$\|w\|_2^2$ reg.

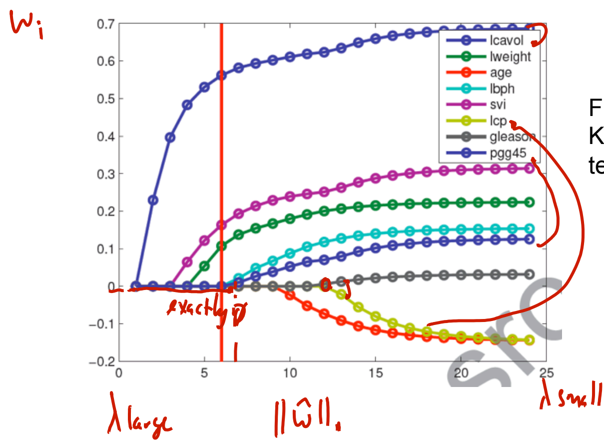
w_i



From Kevin Murphy textbook

- Typical approach: select λ using cross validation

Now: *LASSO Coefficient Path*



From
Kevin Murphy
textbook

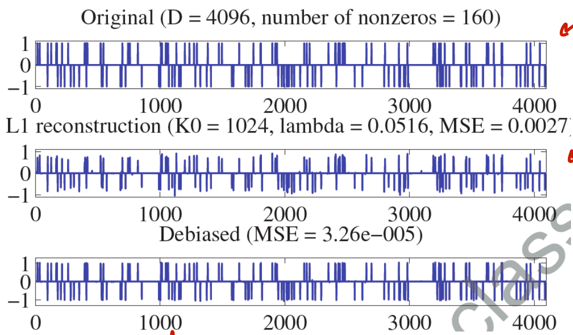
LASSO Example

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



debiasing

LASSO → captures sparsity
↘ but biased toward 0

resolve with unregularized regression only over non-zero $\hat{w}_\ell$

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