

## Simple Variable Selection LASSO: Sparse Regression

Machine Learning – CSE546  
Sham Kakade  
University of Washington

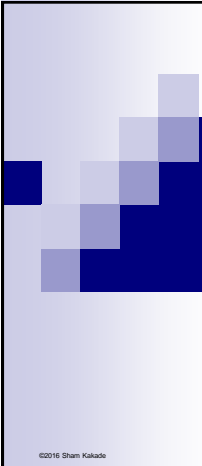
October 11, 2016

©2016 Sham Kakade

### Announcements:

- HW1 due on Friday.
- Readings: please do them.
- Project Proposals: please start thinking about it!
- Today:
  - Review: cross validation
  - Feature selection
  - Lasso

©2016 Sham Kakade



## Review: Cross-Validation

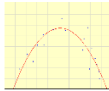
Machine Learning – CSE546  
Sham Kakade  
University of Washington


October 6, 2016

©2016 Sham Kakade

### Regularization in Regression

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

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


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

- **Regularization:** or “Shrinkage” procedure
 
$$\hat{w}_{ridge} = \arg \min_w \sum_{j=1}^N \left( t(x_j) - \left( w_0 + \sum_{i=1}^k w_i h_i(x_j) \right) \right)^2 + \lambda \sum_{i=1}^k w_i^2$$
- How do we pick the regularization constant  $\lambda$ ? (and pick models?)
  - We could use the test set? Or another hold out set?

©2016 Sham Kakade



# Sparsity

*activity of pixels → target*

- Vector  $w$  is sparse, if many entries are zero:
  - $w = [9, 1, 0, 0, \dots, 8, 3, 0, \dots]$
- Very useful for many tasks, e.g.,
  - Efficiency:** If  $\text{size}(w) = 100B$ , each prediction is expensive:
    - 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?
- But computationally intractable to perform "all subsets" regression

Participant P1

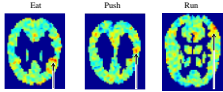


Figure from Tom Mitchell

Mean of independently learned signatures over all nine participants

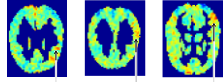


Figure from Tom Mitchell

Pars opercularis (x=24mm)

Postcentral gyrus (x=30mm)

Superior temporal sulcus (posterior) (x=12mm)

©2016 Sham Kalade 9

# Simple greedy model selection algorithm

- Pick a dictionary of features  $\{h_1(x), \dots, h_d(x)\}$ 
  - e.g., polynomials for linear regression
- Greedy heuristic:
  - Start from empty (or simple) set of features  $F_0 = \emptyset$
  - Run learning algorithm for current set of features  $F_t$ 
    - Obtain weights for these features  $\vec{w}_t$
  - Select **next best feature**  $h_i(x)^*$ 
    - e.g.,  $h_i(x)$  that results in lowest training error learner when using  $F_t + \{h_i(x)^*\}$
  - $F_{t+1} \leftarrow F_t + \{h_i(x)^*\}$
  - Recurse

©2016 Sham Kalade 10

# Greedy model selection

- Applicable in many other settings:
  - Considered later in the course:
    - Logistic regression: Selecting features (basis functions)
    - Naive Bayes: Selecting (independent) features  $P(X_i|Y)$
    - Decision trees: Selecting leaves to expand
- Only a heuristic!
  - Finding the best set of  $k$  features is computationally intractable!
  - Sometimes you can prove something strong about it...
- There are many more elaborate methods out there

©2016 Sham Kalade 11

# When do we stop???

- Greedy heuristic:
  - ...
  - Select **next best feature**  $X_i^*$ 
    - E.g.  $h_i(x)$  that results in lowest training error learner when using  $F_t + \{h_i(x)^*\}$
  - Recurse
    - When do you stop???**
      - When training error is low enough?
      - When test set error is low enough?
      - Using cross validation?

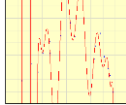
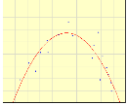
©2016 Sham Kalade 12

## Regularization in Linear Regression

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

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

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



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

## Variable Selection by Regularization

- Ridge regression: Penalizes large weights  $\|w\|_2^2 = \sum_{i=1}^d w_i^2$
- 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

- Try new (**convex**) penalty: Penalize non-zero weights
  - Regularization penalty:

$$\text{Lasso } \|w\|_1 = \sum_i |w_i|$$

- Leads to sparse solutions
- Just like ridge regression, solution is indexed by a continuous param  $\lambda$
- Major impact in: statistics, machine learning & electrical engineering

## LASSO Regression

- LASSO**: least absolute shrinkage and selection operator

- New objective:

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

penalty / regularizer

## (Related) Constrained Optimization

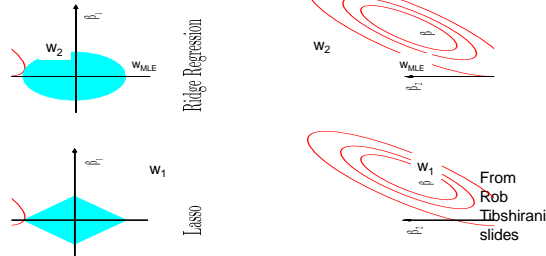
- LASSO solution:

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

Related problem:  $\min_w RSS(w)$  s.t.  $\sum_i |w_i| \leq \beta$

like a  $K$ -sparse

## Geometric Intuition for Sparsity



©2016 Sham Kakade

17

## Optimizing the LASSO Objective

- LASSO solution:

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

$$\frac{\partial F(\mathbf{w})}{\partial w} = 0 \Rightarrow \text{find } \mathbf{w}^*$$

1) What is deriv of  $|w|$

©2016 Sham Kakade

18

## Coordinate Descent

- Given a function  $F$ 
  - Want to find minimum

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} F(w_1, \dots, w_d)$$

- Often, hard to find minimum for all coordinates, but easy for one coordinate
- Coordinate descent: while not converged
  - pick coord.  $l$  randomly
  - $\hat{w}_l \leftarrow \arg \min_w F(\hat{w}_1, \dots, \hat{w}_{l-1}, w, \hat{w}_{l+1}, \dots)$
- How do we pick next coordinate?

- Super useful approach for "many" problems
  - Converges to optimum in some cases, such as LASSO

©2016 Sham Kakade

19

## Optimizing LASSO Objective One Coordinate at a Time

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

- Taking the derivative:

- Residual sum of squares (RSS):

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

- Penalty term:

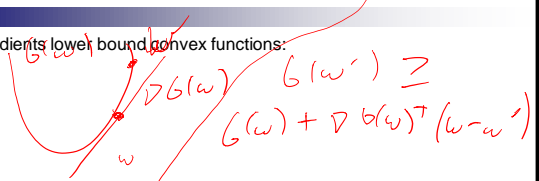
$$\frac{\partial |w|}{\partial w} \quad ??$$

©2016 Sham Kakade

20

## Subgradients of Convex Functions

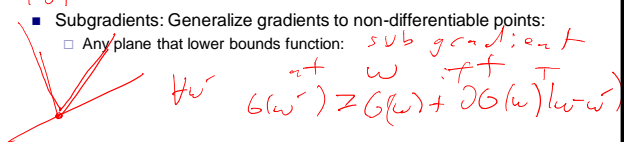
- Gradients lower bound convex functions:



- Gradients are unique at  $\mathbf{w}$  iff function differentiable at  $\mathbf{w}$

- Subgradients: Generalize gradients to non-differentiable points:

- Any plane that lower bounds function: sub gradient:  $\nabla G(w)$



## 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:

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

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

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

- If no penalty:  $w_\ell = c_\ell / a_\ell$
- Subgradient of full objective:

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

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

## Setting Subgradient to 0

$$\frac{\partial F(w)}{\partial w_\ell} = 0$$

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

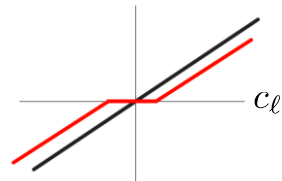
if  $w_\ell < 0$   $\Rightarrow \frac{c_\ell + \lambda}{a_\ell} < 0 \Rightarrow w_\ell = \frac{c_\ell + \lambda}{a_\ell}$  (check  $< 0$ )

if  $w_\ell > 0$   $\Rightarrow \frac{c_\ell - \lambda}{a_\ell} > 0 \Rightarrow w_\ell = \frac{c_\ell - \lambda}{a_\ell}$  (check  $> 0$ )

if  $-\lambda \leq c_\ell \leq \lambda \Rightarrow w_\ell = 0$

## Soft Thresholding

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



From Kevin Murphy textbook

## Coordinate Descent for LASSO (aka Shooting Algorithm)

- Repeat until convergence

- Pick a coordinate  $l$  at (random or sequentially)

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

- Where:

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

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

- For convergence rates, see Shalev-Shwartz and Tewari 2009

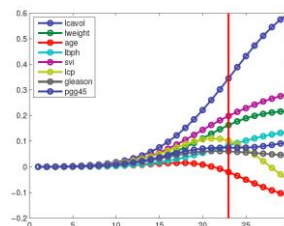
- Other common technique = LARS

- Least angle regression and shrinkage, Efron et al. 2004

©2016 Sham Kalade

25

## Recall: Ridge Coefficient Path



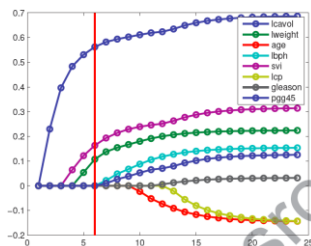
From Kevin Murphy textbook

- Typical approach: select  $\lambda$  using cross validation

©2016 Sham Kalade

26

## Now: LASSO Coefficient Path



From Kevin Murphy textbook

©2016 Sham Kalade

27

## 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 regression
  - 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 simple approach for solving LASSO

©2016 Sham Kalade

28