Regularization in Linear Regression

- Overfitting usually leads to very large parameter choices, e.g.:
  \[-2.2 + 3.1 X - 0.30 X^2\]  
  \[-1.1 + 4,700,910.7 X - 8,585,638.4 X^2 + \ldots\]

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

- $L_2$ regularization
  - regularizes trends
  - smooths functions
Variable Selection

- 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  
  - Computationally intractable to perform “all subsets” regression
  
  \[ 2^K \text{ Subsets to explore} \]

- Try new penalty: Penalize non-zero weights  
  - Regularization penalty:  
  \[ \| w \|_1 = \sum_i |w_i| \]
  - Leads to sparse solutions: Many \( w_i = 0 \)
  - Just like ridge regression, solution is indexed by a continuous param \( \lambda \)
  - This simple approach has changed statistics, machine learning & electrical engineering

LASSO Regression

- **LASSO**: least absolute shrinkage and selection operator
- New objective:
  \[
  \min_w \sum_{j=1}^N \left( f(x_j) - (w_0 + \sum_i w_i x_{ij}) \right)^2 + \lambda \sum_{i=1}^k |w_i|
  \]
  - Don’t regularize \( w_0 \)
Geometric Intuition for Sparsity

From Rob Tibshirani slides

Optimizing the LASSO Objective

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

From quarter thus far: take derivative and set to 0

1. Derivative of |w|? ...

2. Even if you could take derivative, no closed form solution.
Coordinate Descent

- Given a function $F(w)$
  - Want to find minimum
    $$\hat{w} = \arg\min_w F(w)$$
- Often, hard to find minimum for all coordinates, but easy for one coordinate
  - 1-d optimization often much easier
- Coordinate descent: initialize $\hat{w} = \emptyset$
  - While not converged:
    - Pick a coordinate $l$
      $$\hat{w}_l = \arg\min_w F(\hat{w}_0, \hat{w}_1, ..., \hat{w}_{l-1}, w_l, \hat{w}_{l+1}, ..., \hat{w}_K)$$
- How do we pick next coordinate?
  - Round robin, randomly, “smartly”
- Super useful approach for *many* problems
  - Converges to optimum in some cases, such as LASSO

Optimizing LASSO Objective
One Coordinate at a Time

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

- Taking the derivative:
  - Residual sum of squares (RSS):
    $$\frac{\partial}{\partial w_k} RSS(w) = -2 \sum_{j=1}^{N} h_k(x_j) \left( t(x_j) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)$$
  - Penalty term:
    - $|w|$ derivative $= -1$ at zero, undefined otherwise
    - $w$ derivative $= 1$ at zero, undefined otherwise
Subgradients of Convex Functions

- Gradients lower bound convex functions:
  \[ f(w') \geq f(w) + \nabla f(w)'(w' - w) \]

- Gradients are unique at \( w \) iff function differentiable at \( w \)

- Subgradients: Generalize gradients to non-differentiable points:
  - Any plane that lower bounds function:
    \[ \text{For } w \text{ at } 0: \]
    \[ \nabla f(w) \subseteq \{ v \mid v \in [-1, 1] \} \]
    \[ f(w') \geq f(w) + v(w' - w) \]

Taking the Subgradient

- Gradient of RSS term:
  \[ a_\ell = 2 \sum_{j=1}^{N} (h_\ell(x_j))^2 \]
  \[ c_\ell = 2 \sum_{j=1}^{N} h_\ell(x_j) \left( l(x_j) - (w_0 + \sum_{i=1}^{\ell} w_i h_i(x_j)) \right) \]

  - If no penalty: \( \frac{\partial}{\partial w \ell} \text{RSS}(w) = 0 \Rightarrow w = c_\ell / a_\ell \)

- Subgradient of full objective:
  \[ \frac{\partial}{\partial w} f(w) = a_\ell w - c_\ell + \lambda \frac{\partial}{\partial w} \text{I} |w| \]

  - \( \begin{cases} \text{if } w < 0 \\ \text{if } w = 0 \\ \text{if } w > 0 \end{cases} \)

  \[ = \begin{cases} a_\ell w - c_\ell - \lambda \\ [-c_\ell - \lambda, -c_\ell + \lambda] \\ a_\ell w - c_\ell + \lambda \end{cases} \]
Setting Subgradient to 0

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

when \( w_\ell < 0 \) ? \( \Rightarrow \ a_\ell w_\ell - c_\ell - \lambda = 0 \)
\( \Rightarrow \ w_\ell = \frac{c_\ell + \lambda}{a_\ell} \) \( < 0 \Rightarrow \ c_\ell < -\lambda \)

when \( w_\ell > 0 \) ? \( \Rightarrow \ a_\ell w_\ell - c_\ell + \lambda = 0 \)
\( \Rightarrow \ w_\ell = \frac{c_\ell - \lambda}{a_\ell} > 0 \Rightarrow \ c_\ell > \lambda \)

when \( w_\ell = 0 \) ? \( \Rightarrow \ 0 \in [c_\ell - \lambda, -c_\ell + \lambda] \)
\( \Rightarrow \ -\lambda < c_\ell < \lambda \)

Soft Thresholding

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

From Kevin Murphy textbook
Coordinate Descent for LASSO
(aka Shooting Algorithm)

- Repeat until convergence
  - Pick a coordinate $l$ at (random or sequentially)
    - Set:
      \[ \tilde{w}_l = \begin{cases} 
        (c_l + \lambda)/a_l & c_l < -\lambda \\
        0 & c_l \in [-\lambda, \lambda] \\
        (c_l - \lambda)/a_l & c_l > \lambda 
      \end{cases} \]
    - Where:
      \[
      a_l = 2 \sum_{j=1}^{N} (h_l(x_j))^2 \\
      c_l = 2 \sum_{j=1}^{N} h_l(x_j) \left(t(x_j) - (w_0 + \sum_{i \neq l} w_i h_i(x_j))\right)
      \]

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

- Other common technique = LARS

Recall: Ridge Coefficient Path

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

From Kevin Murphy textbook
Now: LASSO Coefficient Path

From Kevin Murphy textbook

LASSO Example

<table>
<thead>
<tr>
<th>Term</th>
<th>Least Squares</th>
<th>Ridge</th>
<th>Lasso</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>2.465</td>
<td>2.452</td>
<td>2.468</td>
</tr>
<tr>
<td>lcavol</td>
<td>0.680</td>
<td>0.420</td>
<td>0.533</td>
</tr>
<tr>
<td>lweight</td>
<td>0.263</td>
<td>0.238</td>
<td>0.169</td>
</tr>
<tr>
<td>age</td>
<td>-0.141</td>
<td>-0.046</td>
<td></td>
</tr>
<tr>
<td>lbph</td>
<td>0.210</td>
<td>0.162</td>
<td>0.002</td>
</tr>
<tr>
<td>svi</td>
<td>0.305</td>
<td>0.227</td>
<td>0.094</td>
</tr>
<tr>
<td>lcp</td>
<td>-0.288</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>gleason</td>
<td>-0.021</td>
<td>0.040</td>
<td></td>
</tr>
<tr>
<td>pgg45</td>
<td>0.267</td>
<td>0.133</td>
<td></td>
</tr>
</tbody>
</table>

From Rob Tibshirani slides

©2005-2013 Carlos Guestrin
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