Feedback

◆ Please complete the surveys!!!
◆ Pods – should be place to get your questions answered
◆ People who answered are spending 10-15 hour/week
  ● I’ll try to dial back the HW
  ● If you’re over 15 hours/week try office hours
◆ Individual comments
  ● Disconnect between the lectures and HW
  ● Would like more ML algorithms (not just theory)
Regression: Penalties & Priors

Learning objectives
Know names and properties of $L_0$, $L_1$ and $L_2$ penalties

Know streamwise, stepwise and stagewise search in regression

Lyle Ungar
Supervised learning

- Given a set of observations with labels, $y$
  - Web pages with “Paris” labeled “Paris, France” or “Paris Hilton”
  - Proteins labeled “apoptosis” or “signaling”
  - Patients labeled with “Alzheimer’s” or “frontotemporal dementia”

- Generate features, $x$, for each observation

- Learn a regression model to predict $y$
  - $y = f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 \ldots$
  - Most of the $w_j$ are zero.
Two interpretations of regression

- Minimize (penalized) squared error
- Maximize likelihood
  - Ordinary least squares (OLS): MLE
    - Minimizes
      A) bias
      B) variance
      C) bias + variance
Two interpretations of regression

◆ Minimize (penalized) squared error
◆ Maximize likelihood
  ● Ridge regression: MAP
    ■ Minimizes
      A) bias
      B) variance
      C) bias + variance
Ridge regression – Bias/Variance

- Minimize penalized Error $||y - Xw||_2^2 + \lambda ||w||_2^2$
  - Minimizing the first term, representing the training error, reduces
    - A) bias
    - B) variance
    - C) neither
Ridge regression – Bias/Variance

Minimize penalized Error \[ ||y - Xw||_2^2 + \lambda \ ||w||_2^2 \]

- Minimizing the second term, which can be viewed as the amount that the test error is expected to be bigger than the training error reduces
  A) bias
  B) variance
  C) neither
**Different norms, different errors**

\[ y \sim N(w^Tx, \sigma^2) \sim \exp(-||y-w^Tx||_2^2/2\sigma^2) \]
- \( \text{argmax}_w p(D|w) \)  
- \( \text{Err} = ||y - Xw||_2^2 \)

OLS = \( L_2 \) regression

\[ y \sim \exp(-||y-w^Tx||_1/2\sigma^2) \]
- \( \text{argmax}_w p(D|w) \)  
- \( \text{Err} = ||y - Xw||_1 \)

L1 regression

here: \( \text{argmax}_w p(y|w,X) \)
Different norms, different penalties

- Minimize penalized Error \[ ||y - Xw||_2^2 + \lambda \; f(w) \]
  - \[ ||w||_2^2 = \sum_j |w_j|^2 \] \( L_2 \)
  - \[ ||w||_1 = \sum_j |w_j|^1 \] \( L_1 \)
  - \[ ||w||_0 = \sum_j |w_j|^0 \] \( L_0 \)
    - Where \( |w_j|^0 = 0 \) if \( w_j = 0 \) else \( |w_j|^0 = 1 \)

- Note that all of these encourage \( w_j \) to be smaller; i.e., they shrink \( w \).
Feature selection for regression

- Goal: minimize error on a test set
- Approximation: minimize a penalized training set error
  \[ \text{Argmin}_w (\text{Err} + \lambda \|w\|_p^p) \text{ where } \text{Err} = \sum_i (y_i - \sum_j w_j x_{ij})^2 = \|y - Xw\|^2 \]
- Different norms
  - \( p = 2 \) – “ridge regression”
    - Makes all the \( w \)'s a little smaller
  - \( p = 1 \) – “LASSO” or “LARS” (least angle regression)
    - Still convex, but drives some \( w \)'s to zero
  - \( p = 0 \) – “stepwise regression”
    - Requires search

Note the confusion in the names of the optimization method with the objective function.
Different regularization priors

\[ \text{Argmin}_w \| y - Xw \|_2^2 + \lambda \| w \|_p^p \]

- **L_2** \( \| w \|_2^2 \)
  - Gaussian prior: \( p(w) \sim \exp(-\| w \|_2^2/\sigma^2) \)

- **L_1** \( \| w \|_1 \)
  - Laplace prior: roughly \( p(w) \sim \exp(-\| w \|_1/\sigma^2) \)

- **L_0** \( \| w \|_0 \)
  - Spike and slab
If the x’s have been standardized (mean zero, variance 1) then we can visualize the shrinkage:

\[ L_2 = \text{Ridge} \]
\[ \text{sum of squares} \]
\[ \text{Shrunken } w \]
\[ w \]

\[ L_1 = \text{Lasso} \]
\[ \text{sum of abs value} \]
\[ \text{Shrunken } w \]
\[ w \]

\[ L_0 = \text{“stepwise regression”} \]
\[ \text{Number of features} \]
\[ \text{Shrunken } w \]
\[ w \]
Different regularization penalties

a) L_2

\[ \text{Argmin}_w \ \|y - Xw\|_2^2 + \lambda \|w\|_2^2 \]

b) L_1

\[ \text{Argmin}_w \ \|y - Xw\|_2^2 + \lambda \|w\|_1 \]

c) L_0

\[ \text{Argmin}_w \ \|y - Xw\|_2^2 + \lambda \|w\|_0 \]

Which norm most heavily shrinks large weights?
Different regularization penalties

a) $L_2$
\[ \text{Argmin}_w \| y - Xw \|^2_2 + \lambda \| w \|^2_2 \]

b) $L_1$
\[ \text{Argmin}_w \| y - Xw \|^2_2 + \lambda \| w \|^1_1 \]

c) $L_0$
\[ \text{Argmin}_w \| y - Xw \|^2_2 + \lambda \| w \|^0_0 \]

Which norm most strongly encourages weights to be set to zero?
Different regularization penalties

a) L₂
\[
\text{Argmin}_w \quad ||y - Xw||_2^2 + \lambda ||w||_2^2
\]

b) L₁
\[
\text{Argmin}_w \quad ||y - Xw||_2^2 + \lambda ||w||_1
\]

c) L₀
\[
\text{Argmin}_w \quad ||y - Xw||_2^2 + \lambda ||w||_0
\]

Which norm is scale invariant?
Different regularization penalties

$$\text{Argmin}_w \ ||y - Xw||_2^2 + \lambda \ ||w||_p^p$$

- L₂ - Ridge regression
- L₁ - LASSO or LARS
- L₀ - “stepwise regression”

Which lead to convex optimization problems?

Warning: for p = 0, the above formula is not really right (here and below); it is really $$||y - w \cdot x||_2^2 + \lambda \ ||w||_0$$
Solving with regularization penalties

\[ \text{Argmin}_w \| y - Xw \|_2^2 + \lambda \| w \|_p^p \]

- **L₂**
  - \((X^T X + \lambda I)^{-1} X'y\)

- **L₁**
  - Gradient descent

- **L₀**
  - Search (stepwise or streamwise)

L₁ and L₀ can handle exponentially more features than observations; L₂ cannot.
Streamwise regression

- **Initialize:**
  - model = {},
  - \( \text{Err}_0 = \sum_i (y_i - \theta)^2 + 0 \)
- **For each feature \( x_j \ j=1:p: \)**
  - **Try** adding the feature \( x_j \) to the model
  - **If**
    - \( \text{Err} = \sum_i (y_i - \sum_{j \in \text{model}} w_j x_{ij})^2 + \lambda \| \text{model} \|_0 < \text{Err}_{j-1} \)
    - Accept new model and set \( \text{Err}_j = \text{Err} \)
  - **Else**
    - Keep old model and set \( \text{Err}_j = \text{Err}_{j-1} \) \( \| \text{model} \|_0 = \# \text{ of features in the model} \)
Stepwise regression

- **Initialize:**
  - model = {},
  - Err\textsubscript{old} = \(\sum_i (y_i - 0)^2 + 0\)

- **Repeat (up to \(p\) times)**
  - **Try** adding each feature \(x_k\) to the model
    - Pick the feature that gives the lowest error
    - \(Err = min_k \sum_i (y_i - \sum_{j in \text{model}_k} w_j x_{ij})^2 + \lambda |\text{model}_k|\)
  - **If** \(Err < Err\textsubscript{old}\)
    - Add the feature to the model
    - \(Err\textsubscript{old} = Err\)
  - **Else** Halt
Stagewise regression

- Like stepwise, but at each iteration, keep all of the coefficients $w_j$ from the old model, and just regress the residual $r_i = y_i - \sum_{j \in \text{model}} w_j x_{ij}$ on the new candidate feature $k$.

Later: boosting
How to pick regularization $\lambda$?

- Search over $\lambda$ to minimize the (non-penalized) error on a test set (or cross validation error)
- Or use information theory for $L_0$.
  - MDL: bits to code model + bits to code residual
What you should know

- L₂, L₁, L₀ penalties
  - Names. How they are solved
- Training vs. Testing
  - Penalized error approximates test error
- Streamwise, stepwise, stagewise regression

L₂ + L₁ penalty = “Elastic net”

$$\arg\min_w ||y - Xw||^2_2 + \lambda_2 ||w||^2_2 + \lambda_1 ||w||_1$$