Regression:
Penalties & Priors

Lyle Ungar
Supervised learning

- Given a set of observations with labels, $y$
  - Observations
    - Web pages with “Paris” labeled “Paris, France” or “Paris Hilton”
    - Proteins labeled “apoptosis” or “signaling”
    - Patients labeled with “alzheimer’s” or “frontoteporal 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 \(|y - w \cdot x|_2^2 + \lambda |w|_2^2\)
  - Minimizing the first term, representing the training error, reduces
    A) bias
    B) variance
    C) neither
Ridge regression – Bias/Variance

- Minimize penalized Error  \( |y - w \cdot x|^2 + \lambda |w|^2 \)
  - 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 penalties

◆ Minimize penalized Error \( |y - w \cdot x|^2 + \lambda f(w) \)

- \( |w|^2 = \sum_j |w_j|^2 \) \( L_2 \)
- \( |w|_1 = \sum_j |w_j|^1 \) \( L_1 \)
- \( |w|_0 = \sum_j |w_j|^0 \) \( L_0 \)
  - 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
  
  \[
  \text{Argmin}_w (\text{Err} + \lambda |w|^p) \quad \text{where} \quad \text{Err} = \sum_i (y_i - \sum_j w_j x_{ij})^2 = \|y - w^T x\|^2
  \]

  - **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”
      - the number of nonzero \( w \)'s

Note the confusion in the names of the optimization method with the objective function.
**Argmin**

$w (\text{Err} + \lambda |w|_p^p)$

- How to pick $\lambda$?

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

- $|w|_2^2 = \Sigma_j |w_j|^2$
- $|w|_1 = \Sigma_j |w_j|^1$
- $|w|_0 = \Sigma_j |w_j|^0$
  - 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 $w$.

- They can be interpreted as different priors on the sizes of the $w$'s
  - E.g. $|w|_2$ is from a Gaussian prior
Different regularization priors

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

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

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

- **L_0**
  - Spike and slab
Different regularization penalties

a) $L_2$

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

b) $L_1$

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

c) $L_0$

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

Which norm most heavily shrinks large weights?
Different regularization penalties

a) $L_2$

$$\text{Argmin}_w \ |y - w \cdot x|^2_2 + \lambda |w|^2_2$$

b) $L_1$

$$\text{Argmin}_w \ |y - w \cdot x|^2_2 + \lambda |w|^1$$

c) $L_0$

$$\text{Argmin}_w \ |y - w \cdot x|^2_2 + \lambda |w|^0$$

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

a) $L_2$

$$\text{Argmin}_w \ |y - w \cdot x|_2^2 + \lambda |w|_2^2$$

b) $L_1$

$$\text{Argmin}_w \ |y - w \cdot x|_2^2 + \lambda |w|_1$$

c) $L_0$

$$\text{Argmin}_w \ |y - w \cdot x|_2^2 + \lambda |w|_0$$

Which norm never forces any features to zero?
Different regularization penalties

a) $L_2$
$$\arg\min_w |y - w \cdot x|_2^2 + \lambda |w|_2^2$$

b) $L_1$
$$\arg\min_w |y - w \cdot x|_2^2 + \lambda |w|_1$$

c) $L_0$
$$\arg\min_w |y - w \cdot x|_2^2 + \lambda |w|_0$$

Which norm is scale invariant?
Different regularization penalties

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

- **L_2** - Ridge regression
- **L_1** - LASSO or LARS
- **L_0** - “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

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

- **L₂**
  - $$(X'X + \lambda I)^{-1} X'y$$

- **L₁**
  - Convex optimization

- **L₁**
  - Search (stepwise or streamwise)

**L₁ and L₀ can handle exponentially more features than observations; L₂ cannot**
If the x’s have been standardized (mean zero, variance 1) then we can visualize the shrinkage:

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

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

\[ L_0 = \text{“stepwise regression”} \]
Number of features
How to pick $\lambda$?

- Try a bunch of different values and see which one minimizes the (non-penalized) error on a test set
  - Cross validation
- Or use information theory for $L_0$. 
Streamwise regression

- **Initialize:**
  - model = {},
  - $Err_0 = \sum_i (y_i - 0)^2 + 0$

- **For each feature $x_j$:**
  - *Try* adding the feature $x_j$ to the model
  - *If*
    - $Err = \sum_i (y_i - \sum_{j \text{ in model}} w_j x_{ij})^2 + \lambda |\text{model}| < Err_{j-1}$
    - Accept new model and set $Err_j = Err$
  - *Else*
    - Keep old model and set $Err_j = Err_{j-1}$

$|\text{model}| = \# \text{ of features in the model}$
Stepwise regression

◆ Initialize:
  - model = {},
  - $\text{Err}_{\text{old}} = \sum_i (y_i - \theta)^2 + 0$

◆ Repeat (up to $p$ times)
  - **Try** adding each feature $x_k$ to the model
    - Pick the feature that gives the lowest error
    - $\text{Err} = \min_k \sum_i (y_i - \sum_{j \in \text{model}_k} w_j x_{ij})^2 + \lambda |\text{model}_k|$
  - **If** $\text{Err} < \text{Err}_{\text{old}}$
    - Add the feature to the model
    - $\text{Err}_{\text{old}} = \text{Err}$
  - **Else** Halt
Like stepwise, but at each iteration, keep all of the coefficients $w_j$ from the old model, and just regress the residual $r_i = y_i - \sum_{j \text{ in model}} w_j x_{ij}$ on the new candidate feature $k$.

Later: boosting
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

\[ \text{L}_2 + \text{L}_1 \text{ penalty} = \text{“Elastic net”} \]

\[ \arg\min_w |y - w \cdot x|^2 + \lambda_2 |w|^2 + \lambda_1 |w|_1 \]
Loss functions

- Is this loss function a distance (metric)?
- Could I use it as the loss function for regression?

\[ y - \hat{y} \]

Error
Distance function (metric)

1. \( d(x, y) \geq 0 \)  (non-negativity, or separation axiom)
2. \( d(x, y) = 0 \) if and only if \( x = y \)  (coincidence axiom)
3. \( d(x, y) = d(y, x) \)  (symmetry)
4. \( d(x, z) \leq d(x, y) + d(y, z) \)  (subadditivity / triangle inequality).

https://en.wikipedia.org/wiki/Metric_(mathematics)
KL divergence

What does KL divergence measure?

\[ D_{KL}(P \| Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}. \]

\[ D_{KL}(P \| Q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} \, dx, \]

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence
KL divergence

◆ Is a distance (metric)
   a) T
   b) F
Data partitioning

- If you are dividing up a data set that someone gives you into a training and test set
  
  A) It is better to randomly select the observations into the two subsets
  B) It is better to divide the data so that the first half is the training set and the second half is the testing set
  C) It is unlikely to matter which one you do