Feedback

- **Please see lectures before your pod**
  - And take the quizzes
- **Do the quizzes surveys**
  - We're being relaxed on due dates, but will become less relaxed
- **People who answered are spending 10-15 hour/week**
  - If you're over 10 hours/week try office hours
- **Lectures/quizzes and worksheets/HW**
  - Theory and practice

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**
  - $y = f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 \ldots$
  - *Most of the w_j are zero.*

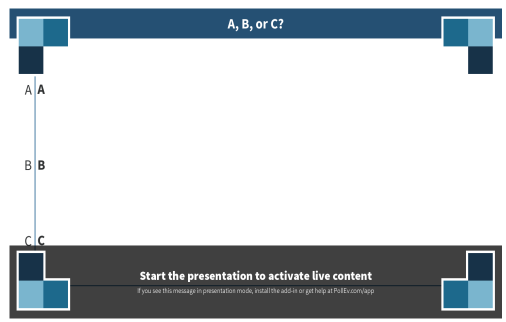
Two interpretations of regression

- **Minimize (penalized) squared error**
- **Maximize likelihood**
    - Ordinary least squares (OLS): MLE
        - Minimizes

          A) bias

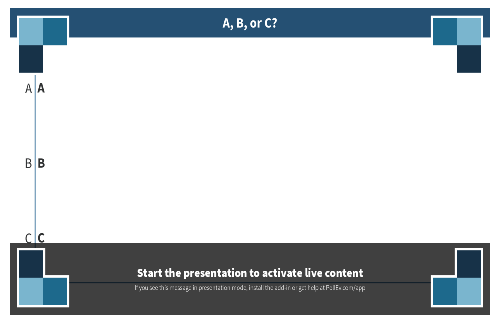
          B) variance

          C) bias + variance



Two interpretations of regression

- **Minimize (penalized) squared error**
- **Maximize likelihood**
  - Ridge regression: MAP
    - Minimizes
      - A) bias
      - B) variance
      - C) bias + variance



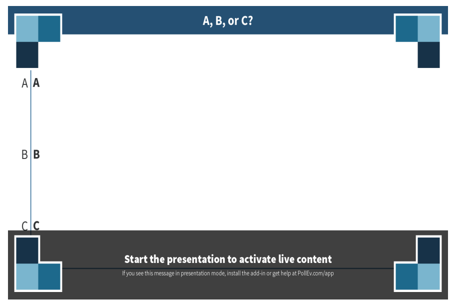
Ridge regression – Bias/Variance

- **Minimize penalized Error** $||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda\ ||\mathbf{w}||_2^2$
  - Minimizing the first term, representing the training error, reduces

    A) bias

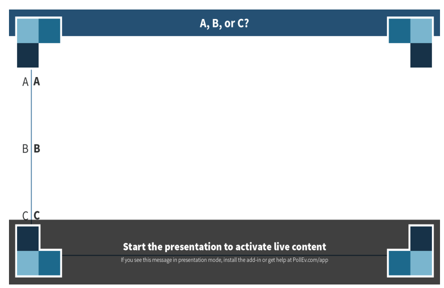
    B) variance

    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

$y \sim N(\mathbf{w}^T\mathbf{x}, \sigma^2) \sim \exp(-||y - \mathbf{w}^T\mathbf{x}||_2^2/2\sigma^2)$

- $\text{argmax}_w\, p(\mathbf{D}|\mathbf{w})$ here: $\text{argmax}_w\, p(\mathbf{y}|\mathbf{w},\mathbf{X})$
- $Err = ||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2$ OLS = L_2 regression

$y \sim \exp(-||y - \mathbf{w}^T\mathbf{x}||_1/2\sigma^2)$

- $\text{argmax}_w\, p(\mathbf{D}|\mathbf{w})$ here: $\text{argmax}_w\, p(\mathbf{y}|\mathbf{w},\mathbf{X})$
- $Err = ||\mathbf{y} - \mathbf{X}\mathbf{w}||_1$ L_1 regression

Different norms, different penalties

- **Minimize penalized Error** $||y - Xw||_2^2 + \lambda\, f(w)$
  - $||w||_2^2 = \Sigma_j\, |w_j|^2$ ***L_2***
  - $||w||_1 = \Sigma_j\, |w_j|^1$ ***L_1***
  - $||w||_0 = \Sigma_j\, |w_j|^0$ ***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.**

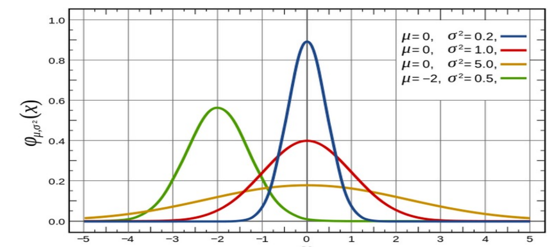
Feature selection for regression

- **Goal: minimize error on a test set**
- **Approximation: minimize a penalized training set error**
  - Argmin$_{\mathbf{w}}$ (Err + λ $||\mathbf{w}||_p^{\,p}$) where Err = $\Sigma_i (y_i - \Sigma_j w_j x_{ij})^2 = ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||^2$
  - **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

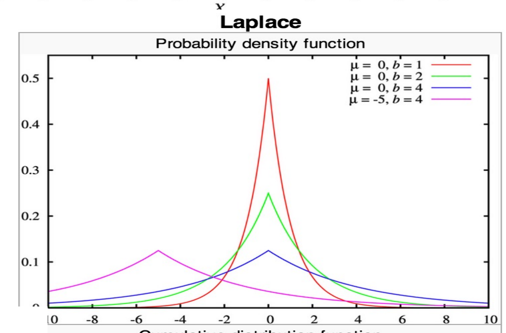
Note the confusion in the names of the of optimization method with the objective function

Different regularization priors

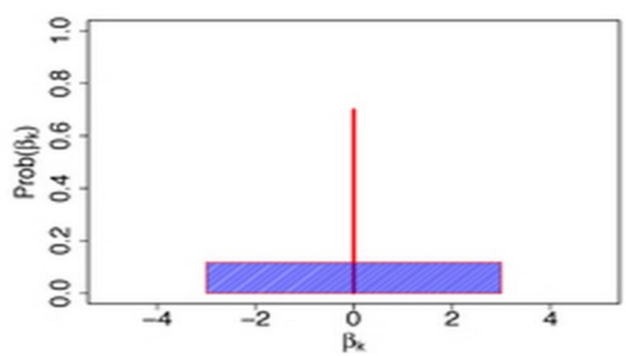
Argmin$_w$ $||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda\, ||\mathbf{w}||_p^p$



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



- **L_1** $||\mathbf{w}||_1$
  - Laplace prior: roughly $p(w) \sim exp(-||\mathbf{w}||_1/\sigma^2)$
- **L_0** $||\mathbf{w}||_0$
  - Spike and slab

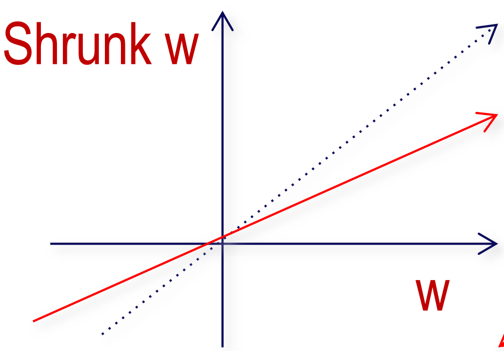


L_0, L_1 and L_2 Penalties

- If *the x's have been standardized* (mean zero, variance 1) then we can visualize the shinkage:

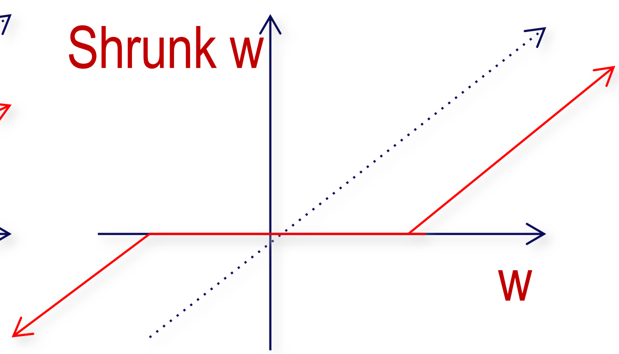
L_2 = Ridge

sum of squares



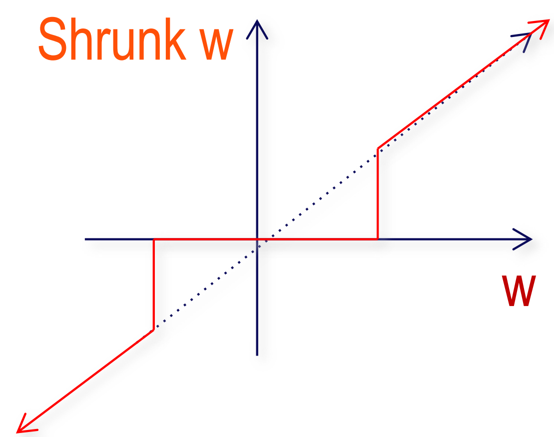
L_1 = Lasso

sum of abs value



L_0 = "stepwise regression"

Number of features



Different regularization penalties

a) L_2

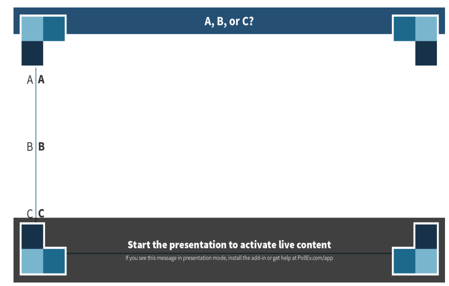
$$\text{Argmin}_w \quad \|\mathbf{y} - \mathbf{Xw}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

b) L_1

$$\text{Argmin}_w \quad \|\mathbf{y} - \mathbf{Xw}\|_2^2 + \lambda \|\mathbf{w}\|_1$$

c) L_0

$$\text{Argmin}_w \quad \|\mathbf{y} - \mathbf{Xw}\|_2^2 + \lambda \|\mathbf{w}\|_0$$



Which norm most heavily shrinks large weights?

Different regularization penalties

a) L_2

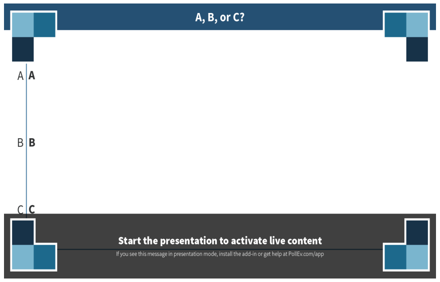
$$\text{Argmin}_w \quad ||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda\, ||\mathbf{w}||_2^2$$

b) L_1

$$\text{Argmin}_w \quad ||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda\, ||\mathbf{w}||_1$$

c) L_0

$$\text{Argmin}_w \quad ||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda\, ||\mathbf{w}||_0$$



Which norm most strongly encourages weights to be set to zero?

Different regularization penalties

a) L_2

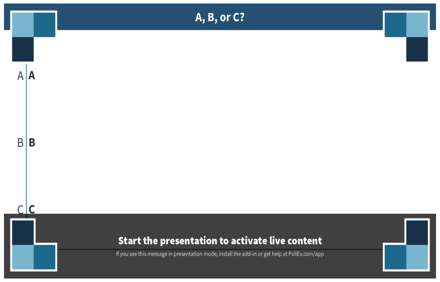
$$\text{Argmin}_w \quad ||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda\ ||\mathbf{w}||_2^2$$

b) L_1

$$\text{Argmin}_w \quad ||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda\ ||\mathbf{w}||_1$$

c) L_0

$$\text{Argmin}_w \quad ||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda\ ||\mathbf{w}||_0$$



Which norm is scale invariant?

Different regularization penalties

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

- **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

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

- **L_2**
  - $(\mathbf{X}^T\mathbf{X} + \lambda\, \mathbf{I})^{-1}\, \mathbf{X'y}$
- **L_1**
  - Gradient descent
- **L_0**
  - Search (stepwise or streamwise)

L_1 and L_0 can handle exponentially more features than observations; L_2 cannot

Streamwise regression

- **Initialize:**
  - model = {},
  - $Err_0 = \Sigma_i (y_i - 0)^2 + 0$
- **For each feature x_j j=1:p:**
  - **Try** adding the feature x_j to the model
  - ***If***
    - $Err = \Sigma_i (y_i - \Sigma_{j \text{ in model}} w_j x_{ij})^2 + \lambda \|model\|_0 < Err_{j-1}$
    - Accept new model and set $Err_j = Err$
  - ***Else***
    - Keep old model and set $Err_j = Err_{j-1}$

$\|model\|_0$ = # of features in the model

Stepwise regression

- **Initialize:**
  - model = {},
  - $Err_{old} = \Sigma_i (y_i - 0)^2 + 0$
- **Repeat (up to *p* times)**
  - **Try** adding each feature x_k to the model
    - Pick the feature that gives the lowest error
    - $Err = min_k \Sigma_i (y_i - \Sigma_{j \, in \, model_k} w_j x_{ij})^2 + \lambda |model_k|$
  - ***If*** $Err < Err_{old}$
    - Add the feature to the model
    - $Err_{old} = Err$
  - ***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* $r_i = y_i - \Sigma_{j \text{ in model}} w_j x_{ij}$ on the new candidate feature $k_.$

Later: boosting

How to pick regularization λ?

- **Search over λ 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

- **L_2, L_1, L_0 penalties**
  - Names. How they are solved
- **Training vs. Testing**
  - Penalized error approximates test error
- **Streamwise, stepwise, stagewise regression**

$L_2 + L_1$ penalty = "Elastic net"

$$\text{argmin}_w \ ||\mathbf{y} - \mathbf{Xw}||_2^2 + \lambda_2 ||\mathbf{w}||_2^2 + \lambda_1 ||\mathbf{w}||_1$$

How is my speed?

Slow

Good

Fast

Start the presentation to activate live content

If you see this message in presentation mode, install the add-in or get help at PollEv.com/app