Midterm Review

2022

Lot of methods!

Different model forms,
Different loss functions
Different optimizations

- **Nonparametric**
  - **Instance-based** (k-nn)
  - **Tree ensembles:** random forest, gradient tree boosting
- **Parametric**
  - Linear, logistic regression, SVM
    - Different loss functions, penalties
    - Equivalence between MLE/MAP and loss/penalized loss

Lots of concepts

- **MLE/MAP**
  - Likelihood to loss function; prior to penalty
- **Entropy, info-gain, KL-divergence**
- **Bias/variance**
- **Norm, distance, kernel**
  - L0, L1, L2
- **Stagewise, residual, boosting**
- **Model forms: linear, logistic, RBF**
- **Regularization methods…**

How to characterize problems

- **n** >> or ~ ore << **p**
- **Is there a clear distance measure between points?**
- **Is feature selection needed?**

For supervised learning!
Later: unsupervised and reinforcement

What to use when?

- **n >> p**
  - Object recognition, speech recognition
- **p >> n**
  - Brain image classification
  - Predict county-level health from twitter word counts
  - Depression from cell phone sensor data
- **p ~ n**
  - Disease diagnosis from medical records

What to use when?

- **n >> p Deep learning**
  - Object recognition, speech recognition
- **p >> n** map to smaller feature space (RBF, PCA)
  - Brain image classification (or K-NN)
  - Predict county-level health from twitter word counts
  - Depression from cell phone sensor data
- **p ~ n Random Forest/GTB, logistic regression**
  - Disease diagnosis from medical records
  - Need to regularize regression

How to regularize (p~n)

- If most features are expected to be correlated with the outcome
- If very few features are expected to be correlated with the outcome

How to regularize (p~n)

- If most features are expected to be correlated with the outcome
  - L1/L2
- If very few features are expected to be correlated with the outcome
  - L0 or L1

Where is the *gradient* in Gradient Tree Boosting?

- **Stagewise updates the model by reducing the Error in the direction of the residual, *r***
  - *d Err/ d r*
- **For an L2 loss, this is accomplished by reducing the residual, since**
  - $d\, Err/ d\, r = d\, ||y - (h_{t-1}(x)+r(x))||_2^2 \sim r$

This allows us to make changes not by changing the weights (as in *d Err/d**w***) in a parametric model, but by adding a correction to the model in the form of a tree

When to standardize data?

- Decision tree?
- k-nn?
- OLS?
- Elastic net?
- L_0 penalized regression?
- SVM?

Where do we use search?

- Decision tree?
- k-nn?
- OLS?
- Elastic net?
- L_0 penalized regression?
- SVM?
- Deep learning?

Kullback Leibler divergence

- ***P*** = true distribution;
- ***Q*** = alternative distribution that is used to encode data
- **KL divergence** is the expected extra message length per datum that must be transmitted using ***Q***

$$
\begin{aligned}
D_{KL}(P \,||\, Q) &= \Sigma_i P(x_i) \log (P(x_i)/Q(x_i)) \\
&= \Sigma_i P(x_i) (-\log Q(x_i) - -\log P(x_i)) \\
&= -\Sigma_i P(x_i) \log Q(x_i) - -\Sigma_i P(x_i) \log P(x_i) \\
&= H(P,Q) - H(P) \\
&= \text{Cross-entropy} - \text{entropy}
\end{aligned}
$$

- **Measures how well Q approximates P**

Where do we use KL-divergence?

- $D(\,p(y \mid x, x') \,\|\, p(y \mid x)\,)$
- $D(\,y \,\|\, h(x)\,)$

Information and friends

- Entropy of the expected value of ____
- KL divergence is the expected value of ___
- Information gain is the difference between ___

Bias Variance Tradeoff

- **Bias:** if you estimate something many times on different training sets, will you systematically be high or low?
- **Variance:** if you estimate something many times on different training sets, how much does your estimate vary?

$$Bias(\hat{\theta}) = E[\hat{\theta} - \theta] = E[\hat{\theta}] - E[\theta]$$
$$Var(\hat{\theta}) = E[(\hat{\theta} - E[\hat{\theta}])^2]$$

Bias Variance Tradeoff - OLS

- Test Error = Variance + Bias^2 + Noise

$$\mathbf{E}_{x,y,D}[(h(x;D)-y)^2] = \underbrace{\mathbf{E}_{x,D}[(h(x;D)-\bar{h}(x))^2]}_{\text{Variance}} + \underbrace{\mathbf{E}_x[(\bar{h}(x)-\bar{y}(x))^2]}_{\text{Bias}^2} + \underbrace{\mathbf{E}_{x,y}[(\bar{y}(x)-y)^2]}_{\text{Noise}}$$

- This applies both to estimating w and to estimating y

$$\text{Error} = E[(y-\hat{y})^2] = Bias(\hat{y})^2 + Var(\hat{y}) + \sigma^2$$

Bias Variance Tradeoff - OLS

- Test Error = Variance + Bias^2 + Noise

$$\mathbf{E}_{x,y,D}[(h(x;D)-y)^2] = \underbrace{\mathbf{E}_{x,D}[(h(x;D)-\bar{h}(x))^2]}_{\text{Variance}} + \underbrace{\mathbf{E}_x[(\bar{h}(x)-\bar{y}(x))^2]}_{\text{Bias}^2} + \underbrace{\mathbf{E}_{x,y}[(\bar{y}(x)-y)^2]}_{\text{Noise}}$$

- This applies both to estimating w and to estimating y

$$\text{Error} = E[(y-\hat{y})^2] = Bias(\hat{y})^2 + Var(\hat{y}) + \sigma^2$$

Bias–Variance Trade-off

Higher complexity = larger or smaller?

bias^2 ____ $\mathbf{E}_x[(\bar{h}(x) - \bar{y}(x))^2]$

variance ____ $\mathbf{E}_{x,D}[(h(x;D) - \bar{h}(x))^2]$

k of k-nn ____

λ of L_p ____

kernel width (RBF) ____

decision tree depth ____

Adaboost

Given: n examples $(\mathbf{x}_i, y_i)$, where $\mathbf{x} \in \mathcal{X}, y \in \pm 1$.

Initialize: $D_1(i) = \frac{1}{n}$

For $t = 1 \ldots T$

- Train weak classifier on distribution $D(i)$, $h_t(\mathbf{x}) : \mathcal{X} \mapsto \pm 1$
- Choose weight α_t (see how below)
- Update: $D_{t+1}(i) = \frac{D_t(i) \exp\{-\alpha_t y_i h_t(\mathbf{x}_i)\}}{Z_t}$, for all i, where $Z_t = \sum_i D_t(i) \exp\{-\alpha_t y_i h_t(\mathbf{x}_i)\}$

Output classifier: $h(\mathbf{x}) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$

Where α_t is the log-odds of the weighted probability of the prediction being wrong

$$\alpha_t = \frac{1}{2}\log \frac{1-\epsilon_t}{\epsilon_t} \qquad \epsilon_t = \sum_i D_t(i)\mathbf{1}(y_i \neq h_t(\mathbf{x_i}))$$

SVM: Hinge loss, ridge penalty

$h(\boldsymbol{x}) = sign(\boldsymbol{w}^T\boldsymbol{x} + b)$

$$\min_{\mathbf{w},b,\xi\geq 0} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\xi_i = \max(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b))$$

0 if score is correct by 1 or more
(hinge loss)

SVM as constrained optimization

Hinge primal:

$$\min_{\mathbf{w},b,\xi\geq 0} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

$$\xi_i = \max(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b))$$

"Slack variable" – hinge loss from the margin

SVM dual

Hinge dual:

$$\max_{\alpha \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \sum_i \alpha_i y_i = 0, \quad \alpha_i \leq C, \quad i = 1, \ldots, n$$

$x_i^T x_j$ is the kernel matrix

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

C controls regularization

Kernel functions k($\mathbf{x}_1$,$\mathbf{x}_2$)

- **Measure similarity or distance?**
- **How to check if something is a kernel function?**
  - Compute a Kernel matrix with elements k($\mathbf{x}_i$,$\mathbf{x}_j$)
  - Make sure its eigenvalues are non-negative
- **Example:** $k(\mathbf{x}_i,\mathbf{x}_j) = x_{i1} + x_{i2} + x_{j1} + x_{j2}$
  - Try the single point $\mathbf{x} = (1,-2)$
  - $K(x,x) = 1-2+1-2 = [-3\ ]$ which is a matrix with eigenvalue -3

When do we need search?

- **Anything with discrete choices**
  - Features in or out of a model
  - Number of hidden nodes or layers
  - Activation function type
- **Most optimization problems (gradient descent)**
  - **Convex:** Logistic regression, SVMs
  - **Nonconvex:** neural nets

Stepwise regression

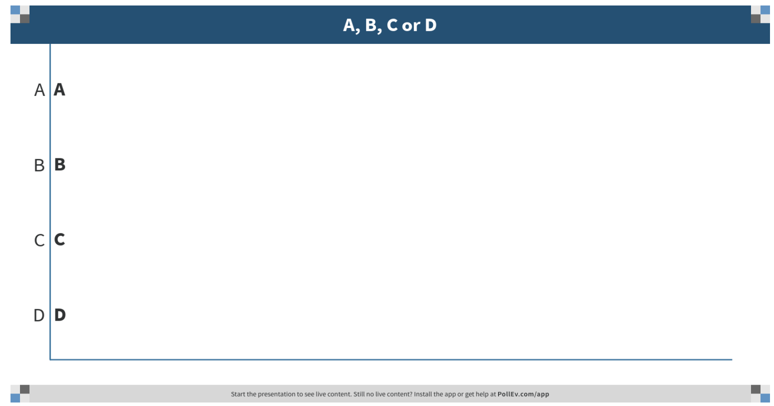
- ***Stepwise regression is used to minimize***

  A) Training set error (MLE)

  B) L_0 penalized training set error

  C) any penalized training set error

  D) None of the above



Why?

Answer: B. If the problem is convex, we don't need stepwise regression

Stepwise regression

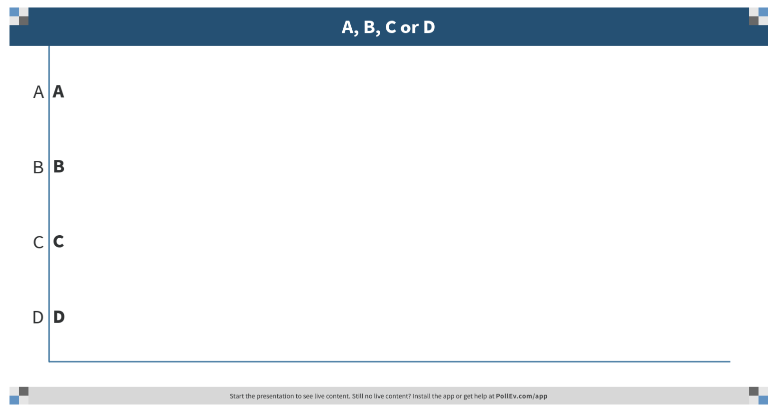
- Given *p* features of which *q* end up being selected
- ***Stepwise regression will estimate …***

  A) q regressions

  B) p regressions

  C) q p regressions

  D) more regressions…



Answer: C.

Streamwise regression

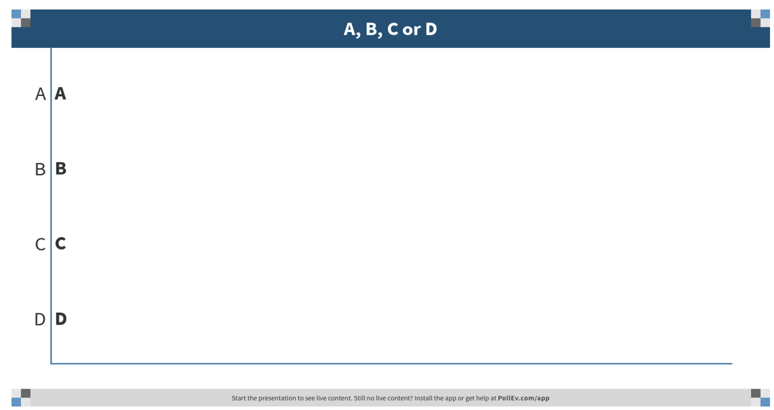
- **Given *p* features of which *q* end up being selected**
- ***Streamwise regression will estimate …***

A) q regressions

B) p regressions

C) q p regressions

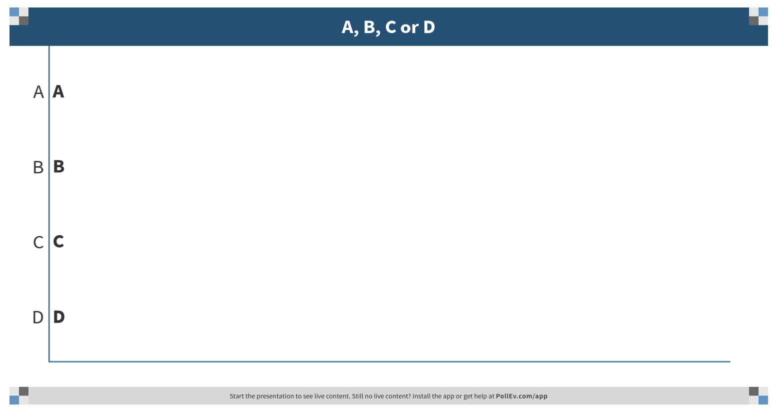
D) more regressions…



Answer: B

Stagewise regression

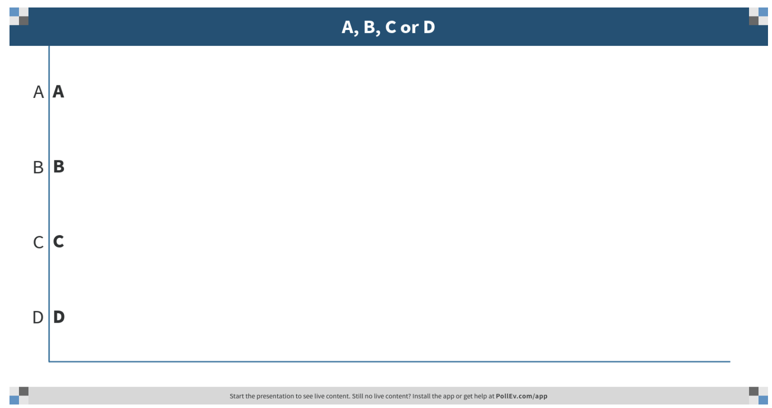
- **Given *p* features of which *q* end up being selected**
- ***Stagewise regression will estimate …***
  - *A) q regressions*
  - *B) p regressions*
  - *C) q p regressions*
  - *D) more regressions…*



Answer: B, if one assumes it is being doing a streamwise search. One could also in theory do a stepwise search, in which case C would be right

Stepwise regression

- **Given *p* features of which *q* end up being selected**
- ***The largest matrix that needs to be inverted is***
  - *A) 1x1*
  - *B) qxq*
  - *C) pxp*
  - *D) bigger*



Answer: B

Stagewise regression

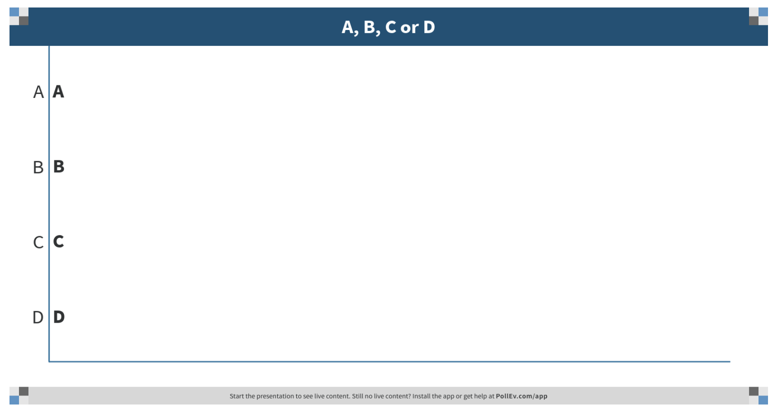
- **Given *p* features of which *q* end up being selected**
- ***The largest matrix that needs to be inverted is***

  A) 1x1

  B) qxq

  C) pxp

  D) bigger



Answer: A we only add one feature at a time!

Streamwise regression - example

- **Assume the true model is**

  $$\mathbf{y = 2\, x_1 + 0\, x_2 + 2\, x_3 + 5\, x_4}$$

  with $\mathbf{x_1 = x_3}$ (two columns are identical)

  and all features standardized

  – thus $\mathbf{x_4}$ will do the most to reduce the error

Streamwise: models are y =

$0,\ 4x_1,\ 4x_1,\ 4x_1\ ,\ 4x_1+5x_4$

Stepwise: models are y =

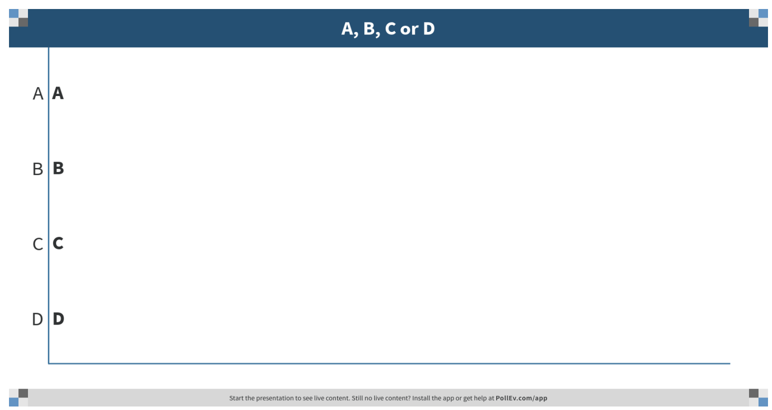
$0,\ 5x_4,\ 4x_1+5x_4$ *or* $4x_3+5x_4$

RBF

- **Transform X to Z using**
  - $z_{ij} = \phi_j(x_i) = k(x_i, \mu_j)$
  - **How many μ_j do we use?**
    - A) $k < p$
    - B) $k = p$
    - C) $k > p$
    - D) any of the above
  - How do we pick k?
  - What other complexity tuner do we have?
- **Linearly regress y on Z**

$$y_i = \Sigma_i a_i \phi_i(\mathbf{x}_i)$$

RBF uses what kernel?



Kernel question

x	y
(1,1)	+1
(1,0)	-1
(0,1)	-1
(-1,1)	+1

Is this linearly separable?

Can you make this linearly separable with 4 Gaussian kernels?

Can you make this linearly separable with 2 Gaussian kernels?

Can you make this linearly separable with 1 Gaussian kernel?

Logistic Regression

$$P(Y=1|\mathbf{x},\mathbf{w}) = \frac{1}{1+\exp\{-\sum_j w_j x_j\}} = \frac{1}{1+\exp\{-\mathbf{w}^\top \mathbf{x}\}} = \frac{1}{1+\exp\{-y\mathbf{w}^\top \mathbf{x}\}}$$

$$P(Y=-1|\mathbf{x},\mathbf{w}) = 1 - P(Y=1|\mathbf{x},\mathbf{w}) = \frac{\exp\{-\mathbf{w}^\top \mathbf{x}\}}{1+\exp\{-\mathbf{w}^\top \mathbf{x}\}} = \frac{1}{1+\exp\{-y\mathbf{w}^\top \mathbf{x}\}}$$

$$log\left(\frac{P(Y=1|\mathbf{x},\mathbf{w})}{P(Y=-1|\mathbf{x},\mathbf{w})}\right) = \mathbf{w}^\top \mathbf{x}$$

Log likelihood of data

$$\log(P(D_Y|D_X,\mathbf{w})) = \log\left(\prod_i \frac{1}{1+\exp\{-y_i\mathbf{w}^\top\mathbf{x}_i\}}\right)$$

$$= -\sum_i \log(1+\exp\{-y_i\mathbf{w}^\top\mathbf{x}_i\})$$

Decision Boundary

$$P(Y=1|\mathbf{x},\mathbf{w}) = P(Y=-1|\mathbf{x},\mathbf{w})$$

$$\frac{1}{1+\exp\{-\mathbf{w}^\top\mathbf{x}\}} = \frac{\exp\{-\mathbf{w}^\top\mathbf{x}\}}{1+\exp\{-\mathbf{w}^\top\mathbf{x}\}}$$

$$\mathbf{w}^\top\mathbf{x} = 0$$

Prediction: $y = \mathrm{sign}(\mathbf{w}^\top\mathbf{x})$

k-class logistic regression

$$P(Y = k|\mathbf{x}, \mathbf{w}) = \frac{\exp\{\mathbf{w}_k^\top \mathbf{x}\}}{\sum_{k'=1}^{K} \exp\{\mathbf{w}_{k'}^\top \mathbf{x}\}}, \quad \text{for} \quad k = 1, \ldots, K$$

Prediction: $y = \text{argmax}_k(\mathbf{w}_k{}^T\mathbf{x})$