What would you most like to see covered on the review session?

"Review of CNN please!!"

"KL-divergence"

"regression penalties"

"Bias&Variance (the error decomposition is very confusing, the wiki page is clear and I thought I understand, but then a bit confused when the homework ask about a similar problem). Others are KL-divergence and boosting."

"SVM"

"Regression, Bias-Variance Decomposition"

"RBFs, regression penalties,"

"Boosting, Decision trees construction (how to split in case of regression having ranges and how does it actually happen)"

"AdaBoost"
Announcements

- Midterm Wednesday
- HW4 due Monday (no extensions)
  - Solutions will be posted
- Please don’t cheat!
Midterm Review

2020
CNN

Input Volume (+pad 1) (7x7x3)

Filter W0 (3x3x3)

Filter W1 (3x3x3)

Output Volume (3x3x2)

Bias b0 (1x1x1)

Bias b1 (1x1x1)
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 $Q$

$$D_{KL}(P \mid\mid Q) = \sum_i P(x_i) \log \left( \frac{P(x_i)}{Q(x_i)} \right)$$

$$= - \sum_i P(x_i) \log Q(x_i) + \sum_i P(x_i) \log P(x_i)$$

$$= H(P,Q) - H(P)$$

$$= \text{Cross-entropy} - \text{entropy}$$

- Measures how different the two distributions are
Where do we use KL-divergence?

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

\[ E_{x,y,D}[(h(x; D) - y)^2] = E_{x,D}[(h(x; D) - \bar{h}(x))^2] + E_x[(\bar{h}(x) - \bar{y}(x))^2] + E_{x,y}[(\bar{y}(x) - y)^2]. \]

\[ \text{Variance} \quad \text{Bias}^2 \quad \text{Noise} \]

\[ \text{This applies both to estimating } w \text{ and to estimating } y \]

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

- Test Error = Variance + Bias\(^2\) + Noise

\[
E_{x,y,D}[(h(x; D) - y)^2] = E_{x,D}[(h(x; D) - \bar{h}(x))^2] + E_x[(\bar{h}(x) - \bar{y}(x))^2] + E_{x,y}[(\bar{y}(x) - y)^2].
\]

- This applies both to estimating w and to estimating y

\[
\text{Error} = E[(y - \hat{y})^2] = \text{Bias}(\hat{y})^2 + \text{Var}(\hat{y}) + \sigma^2.
\]
Bias–Variance Trade-off

Higher complexity = larger or smaller?

$\text{bias}^2$ \quad $E_x[\overline{(\hat{h}(x) - \overline{y}(x))^2}]$

variance \quad $E_{x,D}[\overline{(h(x; D) - \overline{h}(x))^2}]$

$k$ of $k$-nn

$\lambda$ of $L_p$

kernel width (RBF)

? of decision trees
Adaboost

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

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

For \(t = 1 \ldots T\)

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

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

Where \(\alpha_t\) is the log-odds of the weighted probability of the prediction being wrong

\[\alpha_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t} \quad \epsilon_t = \sum_i D_t(i) 1(y_i \neq h_t(x_i))\]
SVM: Hinge loss, ridge penalty

\[ h(x) = \text{sign}(w^T x + b) \]

\[
\min_{w,b,\xi \geq 0} \frac{1}{2} w^T w + C \sum_i \xi_i
\]

\[ \xi_i = \max(0, 1 - y_i(w^T x_i + b)) \]

0 if score is correct by 1 or more (hinge loss)
SVM as constrained optimization

Hinge primal:

\[
\min_{w,b,\xi \geq 0} \frac{1}{2} w^\top w + C \sum_i \xi_i \\
\text{s.t. } y_i (w^\top x_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n
\]

\[
\xi_i = \max(0, 1 - y_i (w^\top x_i + b))
\]

“Slack variable” – hinge loss from the margin
SVM dual

Hinge dual:

\[
\begin{align*}
\max_{\alpha \geq 0} & \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
\text{s.t.} & \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \leq C, \quad i = 1, \ldots, n
\end{align*}
\]

\(x_i^T x_j\) is the kernel matrix

\(w = \sum_i \alpha_i y_i x_i\)

C controls regularization
Scale invariance

- Decision tree?
- k-nn?
- OLS?
- Elastic net?
- $L_0$ penalized regression?
- SVM?
Kernel functions $k(x_1,x_2)$

- Measure similarity or distance?
- How to check if something is a kernel function?
  - Compute a Kernel matrix with elements $k(x_i,x_j)$
  - Make sure its eigenvalues are non-negative

**Example:** $k(x_i,x_j) = x_{i1} + x_{i2} + x_{j1} + x_{j2}$
  - Try the single point $x = (1,-2)$
  - $K(x,x) = 1-2+1-2 = [-3]$ which is a matrix with eigenvalue -3
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?
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…
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…
Stagewise regression

- Given $p$ features of which $q$ end up being selected
- Stagewise regression will estimate ...
  A) $q$ regressions
  B) $p$ regressions
  C) $q \times p$ regressions
  D) more regressions...
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
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
Streamwise regression - example

- Assume the true model is
  \[ y = 2x_1 + 0x_2 + 2x_3 + 5x_4 \]
  with \( x_1 = x_3 \) (two columns are identical)
  and all features standardized
  - thus \( x_4 \) 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 \text{ or } 4x_3 + 5x_4
**RBF**

- **Transform X to Z using**
  - \( z_{ij} = \phi_j(x_i) = k(x_i, \mu_j) \)
  - How many \( \mu_j \) 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**
  \[ y_i = \sum_j a_j \phi_j(x_i) \]
## Kernel question

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1,1)</td>
<td>+1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1,0)</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0,1)</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1,1)</td>
<td>+1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

\[ P(Y = 1|x, w) = \frac{1}{1 + \exp\{-\sum_j w_j x_j\}} = \frac{1}{1 + \exp\{-w^T x\}} = \frac{1}{1 + \exp\{-yw^T x\}} \]

\[ P(Y = -1|x, w) = 1 - P(Y = 1|x, w) = \frac{\exp\{-w^T x\}}{1 + \exp\{-w^T x\}} = \frac{1}{1 + \exp\{-yw^T x\}} \]

\[ \log\left(\frac{P(Y=1|x,w)}{P(Y=-1|x,w)}\right) = w^T x \]
Log likelihood of data

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

$$= - \sum_i \log(1 + \exp\{-y_i w^T x_i\})$$
Decision Boundary

\[ P(Y = 1 | x, w) = P(Y = -1 | x, w) \]

\[ \frac{1}{1 + \exp{-w^T x}} \quad = \quad \frac{\exp{-w^T x}}{1 + \exp{-w^T x}} \]

\[ w^T x = 0 \]

Prediction: \( y = \text{sign}(w^T x) \)
k-class logistic regression

\[ P(Y = k | x, w) = \frac{\exp\{w_k^T x\}}{\sum_{k'=1}^{K} \exp\{w_{k'}^T x\}} , \quad \text{for} \quad k = 1, \ldots, K \]

Prediction: \( y = \arg\max_k (w_k^T x) \)
GANS
Generative Adversarial Networks: GANS

G: \( \text{argmin} \ \log(1 - D(G(\text{noise}))) \)

https://medium.freecodecamp.org/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394
Conditional GANS