Midterm Review
2018
MDL
Bias–Variance Trade-off

Higher complexity = larger or smaller?

\[ \text{bias}^2 \quad \text{variance} \quad k \text{ of \, k-nn} \quad \lambda \text{ of \, } L_p \quad \text{kernel width (RBF)} \quad \text{? \ of \ decision \ trees} \]

\[ \mathbb{E}_x[(\bar{h}(x) - \bar{y}(x))^2] \quad \mathbb{E}_{x,D}[(h(x; D) - \bar{h}(x))^2] \]
Two part code: what are the two parts?

- Residual
- Model

  - Two part code: what are the two parts
    - Which features are in the model?
    - Code the features.
MDL

◆ **Sender and Receiver**
  * What do they know?
  * What do they send?

◆ **Two part code: What are the two parts?**
  * Residual
  * Model
    * Two part code: What are the two parts?
      * Which features are in the model?
      * Code the features.
MDL: Code residual & model

- Residual = bias or variance?
- Model = bias or variance?
Cost to code residual?

- How accurately do we code the SSE?
  - $\sigma^2$
If a feature has a probability of $q/p$ of coming into a model, the expected number of bits to code the presence or absence of the feature is

$$-(q/p)\log(q/p) - (1-q/p)\log(1-q/p)$$

The expected cost of coding all $p$ features is

$$-(q)\log(q/p) - (p-q)\log(1-q/p)$$

If $q=1$, the cost is $\log(p)$ and the penalty is called RIC.
Cost to code which features in model?

- If a feature has a probability of $q/p$ of coming into a model, the expected number of bits to code the presence or absence of the feature is:
  - $-(q/p)\log(q/p) - (1-q/p)\log(1-q/p)$

- If $q=p/2$, the cost per feature is:
  - $-(1/2)\log(1/2)-(1/2)\log(1/2) = \log(2) = 1$
  - or 2 bits per included feature
  - and the method is called
    - AIC
Cost of coding each feature $\sim n^{1/2}$

Which penalty assumes this term dominates?

A) AIC
B) BIC
C) RIC
D) None of the above
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) $q \times p$ 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) $q$ $p$ 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) $1 \times 1$
B) $q \times q$
C) $p \times p$
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
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>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>(1,1)</td>
<td>+1</td>
</tr>
<tr>
<td>(1,0)</td>
<td>-1</td>
</tr>
<tr>
<td>(0,1)</td>
<td>-1</td>
</tr>
<tr>
<td>(-1,1)</td>
<td>+1</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

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

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

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

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

\[ = - \sum_i \log(1 + \exp\{-y_i w^\top x_i\}) \]
Decision Boundary

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

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

\[
w^T x = 0
\]

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: \( y = \arg\max_k (w_k^T x) \)
Naïve Bayes

◆ **Bayes rule**
  - $P(Y=y|X=x) = P(X=x|Y=y) \ p(Y=y) / P(X=x)$
  - Prior, likelihood and posterior

◆ **What assumptions do we make?**
  - $P(Y=y|X=x) \sim P(X=x|Y=y) \ p(Y=y)$
  - $P(Y=y|X=x) \sim P(X_1=x_1|Y=y) \ P(X_2=x_2|Y=y) \cdots P(X_p=x_p|Y=y) \ p(Y=y)$

◆ **MLE or MAP estimation?**

◆ **What extra assumption is made for language?**
Naïve Bayes Example

Data
Y=good  X = “I”, “love”, “math”
Y=good  X = “I”, “love”, “CIS520”
Y=bad   X = “I”, “hate”, “exams”

Estimate \( P(Y=\text{good}|X = \text{“I”, “love”, “exams”}) \)

\[
P(Y=\text{good}|X) \sim P(X|Y=\text{good}) \cdot P(Y=\text{good})
\]
\[
P(Y=\text{bad}|X) \sim P(X|Y=\text{bad}) \cdot P(Y=\text{bad})
\]
Naïve Bayes Example

Data
Y=good  X = “I”, “love”, “math”
Y=good  X = “I”, “love”, “CIS520”
Y=bad   X = “I”, “hate”, “exams”

Estimate Prior
P(Y=good) = _______
Naïve Bayes Example

Data
Y=good  X = “I”, “love”, “math”
Y=good  X = “I”, “love”, “CIS520”
Y=bad   X = “I”, “hate”, “exams”

Estimate likelihood (MLE)
P(X = “I”, “love”, “exams” | Y=good) = ______
P(X=“I” | Y=good) P(X=“love” | Y=good) P(X= “exams” | Y=good)
1                                  1                                     0
Naïve Bayes Example

Data
Y=good  X = “I”, “love”, “math”
Y=good  X = “I”, “love”, “CIS520”
Y=bad   X = “I”, “hate”, “exams”

Estimate likelihood (Laplace smoothing = MAP)
P(X = “I”, “love”, “exams” | Y=good) =
P(X=“I” | Y=good) P(X=“love” | Y=good) P(X= “exams” | Y=good)
1                      1                         1/3
Naïve Bayes Example

Data

Y=good  X = “I”, “love”, “math”
Y=good  X = “I”, “love”, “CIS520”
Y=bad   X = “I”, “hate”, “exams”

Estimate $\arg\max_Y P(Y|X = \text{“I”, “love”, “exams”})$

$P(Y=\text{good}|X) \sim P(X|Y=\text{good}) \ P(Y=\text{good}) = (1 * 1 * 1/3)(2/3)$

$P(Y=\text{bad}|X) \sim P(X|Y=\text{bad}) \ P(Y=\text{bad}) = (1 * 1/3 * 1)(1/3)$
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