Model Selection

CSCC11 – Topic 04

Computer & Mathematical Sciences
UNIVERSITY OF TORONTO
SCARBOROUGH

Defy Gravity • • • Inspiring Exclusive Excellence

Cross Validation

Model Selection

- How do we select hyperparameters

Model	Hyperparameters
K-NN	K
Basis Function Regression	# basis functions, regularization coefficient RBF width and spacing, polynomial degree

- We care about generalization: want the model perform well on unseen data.
- Cross Validation
  - Hold out part of the data as validation data from training
  - Used in statistics for a long time

Hold-out Validation

- Partition data randomly into training set and validation set
- Train on the training set
- Validate (compare models) on the validation set
- Do not use training data to select your hyperparameters
- Advantages
  - Model agnostic
  - Simple conceptually
  - You can use different loss functions in training and validation
    - 0-1 Loss cannot be used in training, but can be used in validation

Using Validation Set to Select Hyperparameter

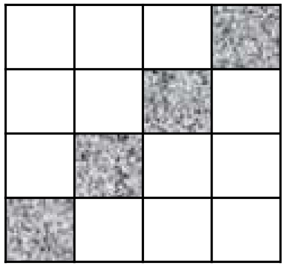
- Partition data into training set, validation set and test set.
- Let hyperparameter $\lambda \in \{\lambda_1, \cdots \lambda_C\}$, train the model for all possible values of λ
- Let Err_λ be the error on the validation set when hyperparameter is set to λ and weights are obtained from the training set.
- Note the test set is for reporting the performance of your model after the hyperparameter is selected and the model is trained.

For λ in $\{\lambda_1, \cdots \lambda_C\}$
    $\mathcal{M}_\lambda \leftarrow$ train(λ, training set)
    $Err_\lambda \leftarrow$ test($\mathcal{M}_\lambda$, validation set)
$\lambda^* \leftarrow \underset{\lambda}{\operatorname{argmin}}\, Err_\lambda$
$\mathcal{M} \leftarrow$ train(λ^*, training set $\cup$ validation set)
$Err \leftarrow$ test($\mathcal{M}$, test data)
Return λ^*, $\mathcal{M}$, Err

K-Fold Cross Validation

- If the dataset is small, then either training or validation set may be too small to be reliable.
- K-Fold Cross Validation
  - Partition data in K subsets
  - For each subset, learn model on the remaining $(k-1)$ subsets
  - Let $Err_{i,\lambda}$ be the error on the i-th subset for the model trained on all other subsets when hyperparameter is λ.
  - Total cross validation error is given by

$$Err_{\lambda} = \frac{1}{K}\sum_{i=1}^{K} Err_{i,\lambda}$$



$K = 4$

K-Fold Cross Validation

```
for λ in {λ_1, ⋯ λ_C}
    for i=1 to K do (i indexes the training set splits)
        M_{i,λ} ← train(λ, training sets {1,…,i-1, i+1,…, K})
        Err_{i,λ} ← test(M_{i,λ}, validation set i)
    Err_λ = (1/K) Σ_{i=1}^{K} Err_{i,λ}
λ* ← argmin_λ Err_λ
M ← train(λ*, training sets {1, …, K})
Err ← test(M, test data)
Return λ*, M, Err
```

$$
\begin{aligned}
&\texttt{for } \lambda \texttt{ in } \{\lambda_1, \cdots \lambda_C\} \\
&\quad \texttt{for i=1 to } K \texttt{ do (i indexes the training set splits)} \\
&\quad\quad \mathcal{M}_{i,\lambda} \leftarrow \texttt{train}(\lambda, \texttt{ training sets } \{1,\ldots,\texttt{i-1, i+1},\ldots, K\}) \\
&\quad\quad Err_{i,\lambda} \leftarrow \texttt{test}(\mathcal{M}_{i,\lambda}, \texttt{ validation set i}) \\
&\quad Err_\lambda = \frac{1}{K}\sum_{i=1}^{K} Err_{i,\lambda} \\
&\lambda^* \leftarrow \underset{\lambda}{\operatorname{argmin}}\, Err_\lambda \\
&\mathcal{M} \leftarrow \texttt{train}(\lambda^*, \texttt{ training sets } \{1, \ldots, \texttt{K}\}) \\
&Err \leftarrow \texttt{test}(\mathcal{M}, \texttt{ test data}) \\
&\texttt{Return } \lambda^*, \mathcal{M}, Err
\end{aligned}
$$

Leave One Out Cross Validation

- LOOCV is a special case when $K = N$
  - Take one data point out as the validation set
  - Train the model on the rest of the data
  - We learn N models
    - When N is big, we have to learn big number of models
- For linear basis function regression with squared loss

$$\text{LOOCV} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

N Models

Prediction from the i^{th} model

LOOCV cont'd

- For Linear basis function regression, we can just learn one model fit.

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

X: design matrix
y: vector of training output
$\hat{\mathbf{y}}$: $\mathbf{X}\mathbf{w}^*$ predicted output on training input

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}^* = \underbrace{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}_{\mathbf{H}}\mathbf{y} = \mathbf{H}\mathbf{y}$$

$$\text{LOOCV} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2$$

h_i is the i-th diagonal entry in $\mathbf{H}$

Problems with Cross Validation

- Computationally expensive
- With m hyperparameters, each has C distinct values to be tested
- We need to learn C^m distinct models
- For K-Fold cross validation, we need to learn KC^m models
- It is good for small number of hyperparameters (1,2 and 3).