

Model Selection

CSCC11 – Topic 04



Computer & Mathematical Sciences
UNIVERSITY OF TORONTO
SCARBOROUGH

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. 泛化性
- 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 所知
 - Simple conceptually 简单概念
 - 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

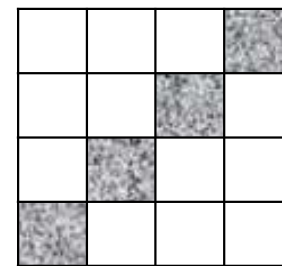
- Partition data into training set, validation set and test set.
- Let hyperparameter $\lambda \in \{\lambda_1, \dots, \lambda_C\}$, train the model for all possible values of λ
- Let Err_λ be the error on the validation set when hyperparameter is set to λ and weights are obtained from the training set.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

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

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

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.

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

\mathbf{X} : design matrix

\mathbf{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).

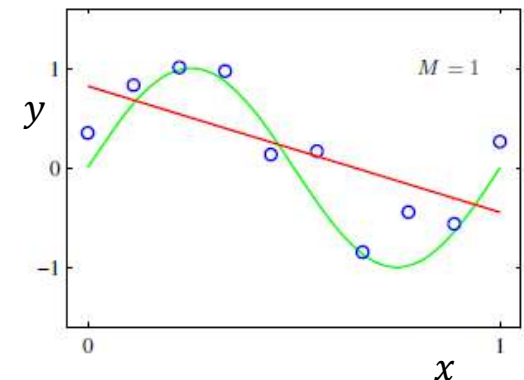
Bias and Variance Decomposition

Bias and Underfitting

- Bias
 - is the model's tendency to make **systematic errors** due to its strict constraints on the data representation.
- Underfitting – High Bias
 - High training error, high test error
 - The model is too simple to capture the structure of the data
 - The model with high bias limits the flexibility to capture the actual pattern in the data.
 - The model generalizes too broadly
 - Even with unlimited data, the model fails to fit the data well.

$$y = f(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$$

$$\hat{y} = \hat{f}(x) = wx + b$$



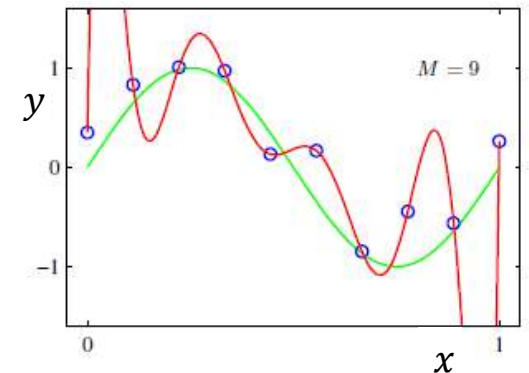
[Image Courtesy of Bishop's PRML]

Variance and Overfitting

- Variance
 - Measures how the model varies when there are changes in the training dataset
 - Quantifies the sensitivity of the model to small fluctuations in the training set
- Overfitting – High Variance
 - Low training error, high test error
 - The model is too complex and captures idiosyncratic characteristics of the particular training set that are not the true structure of the data.
 - If we change the training set slightly, we see the model varies significantly

$$y = f(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$$

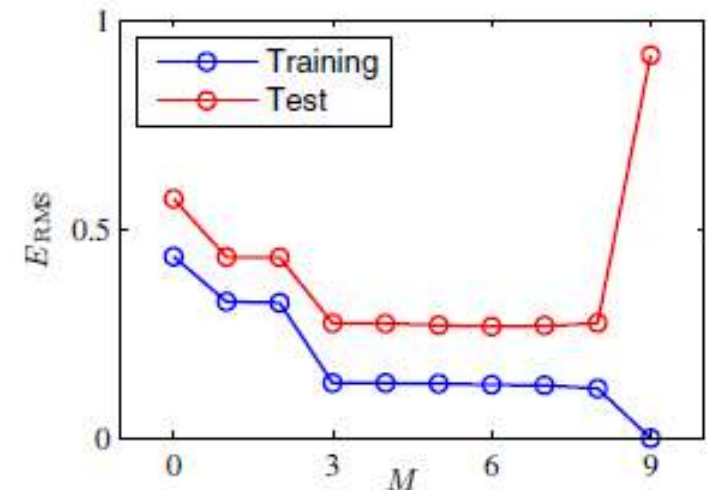
$$\hat{y} = \hat{f}(x) = w_9 x^9 + \dots + w_1 x + b$$



[Image Courtesy of Bishop's PRML]

Generalization Error

- The error of the predictor on the unseen test data quantifies how well our learning algorithm generalizes.
- Source of generalization errors
 - Noises in the test data:
 - nothing can be done by improving the model
 - Errors that are from the model
 - Bias: systematic errors
 - Variance: errors due to change of training set
- Bias variance decomposition is to look overfitting and underfitting problem from a frequentist point of view



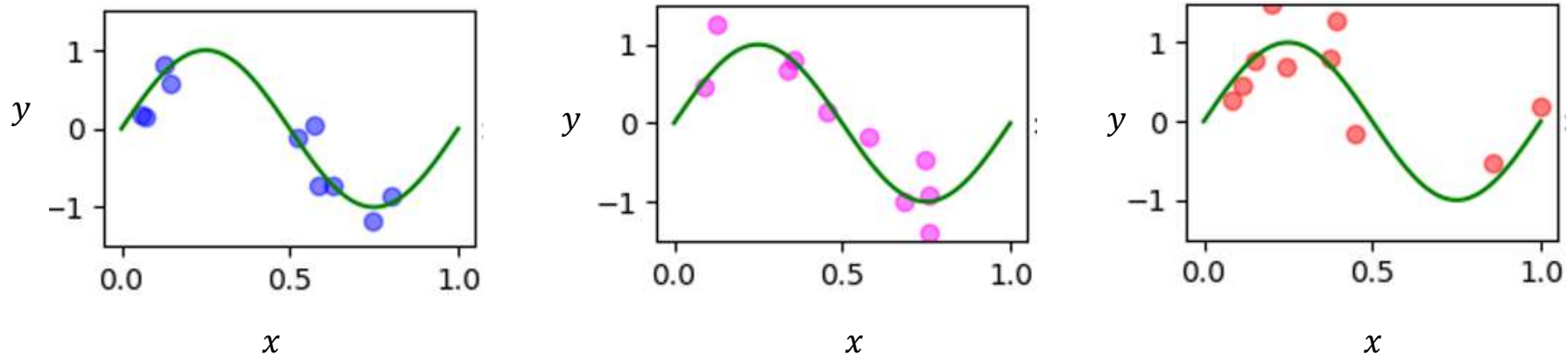
Graphs of the root-mean-square error, defined by (1.3), evaluated on the training set and on an independent test set for various values of M .

[Image Courtesy of Bishop's PRML]

Frequentist vs Bayesian Views

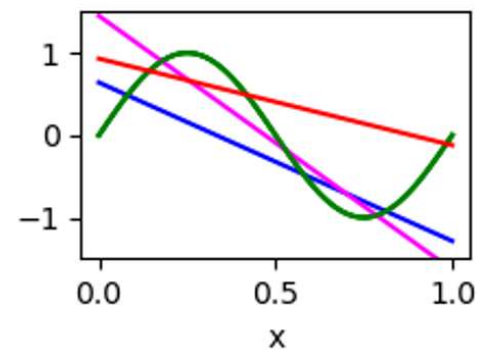
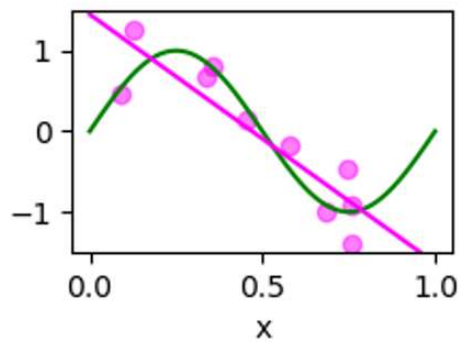
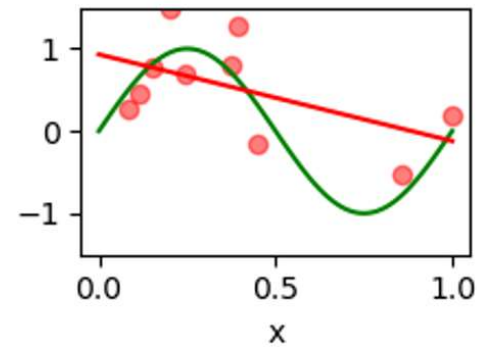
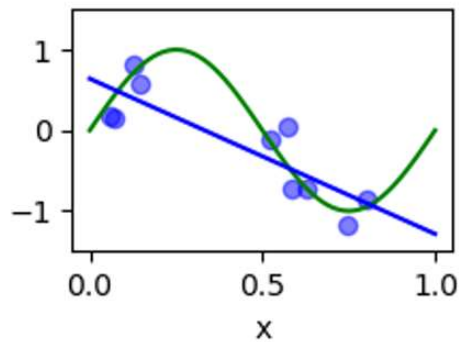
- In the Bayesian methods, we assume the training dataset is fixed and the uncertainty lies in the model parameter
- From a frequentist point view, the training data is a sample from a distribution. We repeat the training on different training sets sampled from the data distribution
 - For each sampled training set, we compute the model parameter \mathbf{w}^* using a learning algorithm \mathcal{A}
 - The predicted output value y is fixed give \mathbf{w}^* , but varies if we vary training sets for the same learning algorithm \mathcal{A}
 - To evaluate how good the predictor is from algorithm \mathcal{A} , we use the average of predicted y values that are learned from different training sets.

Training Set Sampling

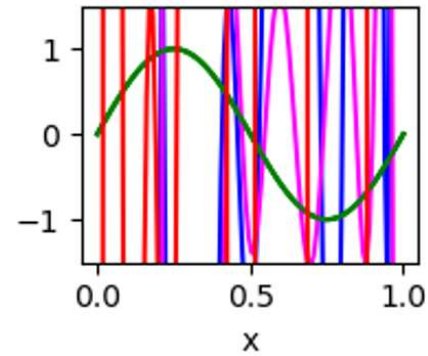
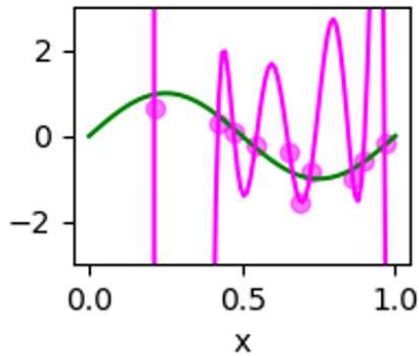
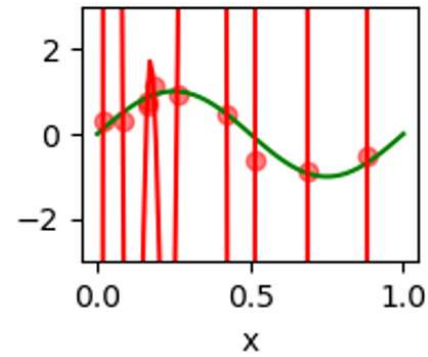
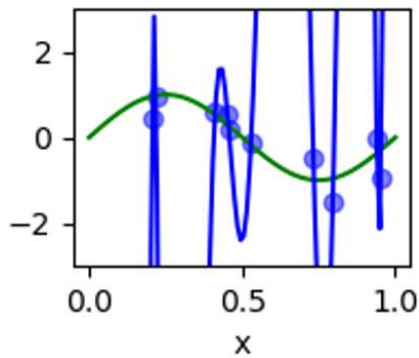


- The input values $\{x_n\}$ are generated uniformly in range (0, 1)
- The corresponding target values $\{y_n\}$ are obtained by first computing the corresponding values of the function $\sin(2\pi x)$, and then adding random noise with a Gaussian distribution having standard deviation 0.3. [CB]

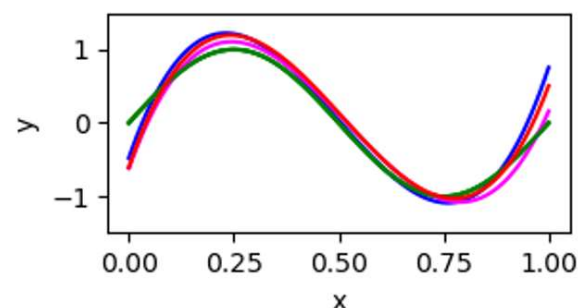
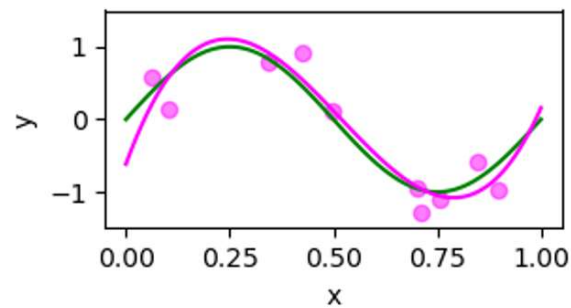
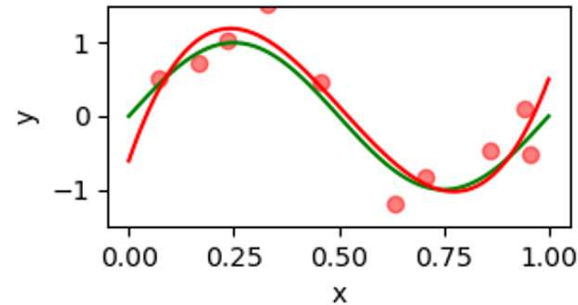
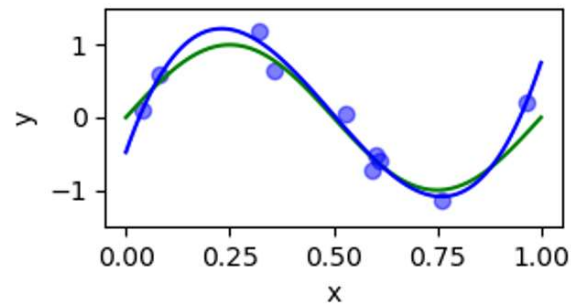
Fitting a Line to Different Training Sets



Fitting a $M=9$ Polynomial to Different Datasets



Fitting a Cubic Polynomial to Different Datasets



Error with a Fixed Training Set \mathcal{D}

- Multiple Regression Model

$$y = \mathbf{w}^T \mathbf{x} + \eta$$

$$y \sim \mathcal{N}(\mathbf{w}^T \mathbf{x}, \sigma^2)$$

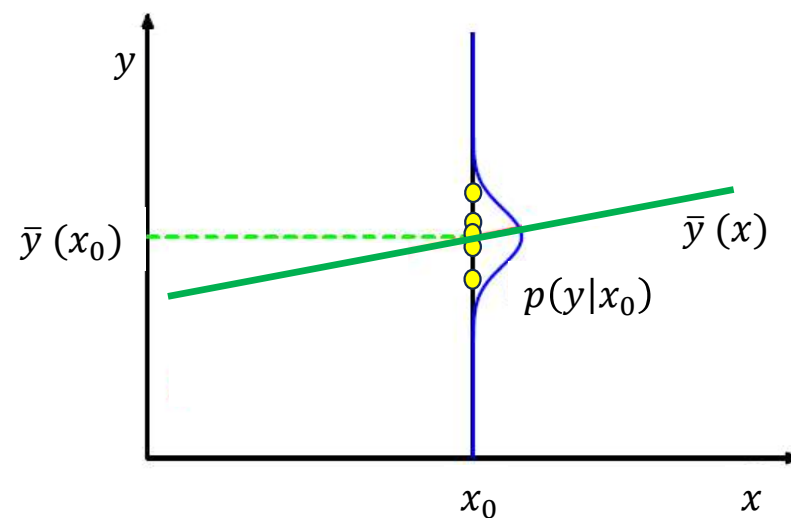
$$\mathbf{w} \in \mathbb{R}^D, \mathbf{x} \in \mathbb{R}^D, \\ y \in \mathbb{R}, \eta \sim \mathcal{N}(0, \sigma^2)$$

Noisy Data

Index	MATB41 x	C11 y
1	90	85
2	71	68
3	90	87
4	60	61
5	90	90
6	60	45

- Given a test point (\mathbf{x}, y) , with the same values of \mathbf{x} , y may have different values. This is due to the noise of the data.
- The expected target value (i.e. label) is

$$\bar{y}(\mathbf{x}) = E_{Y|X}[Y] = \int y p(y|\mathbf{x}) dy$$



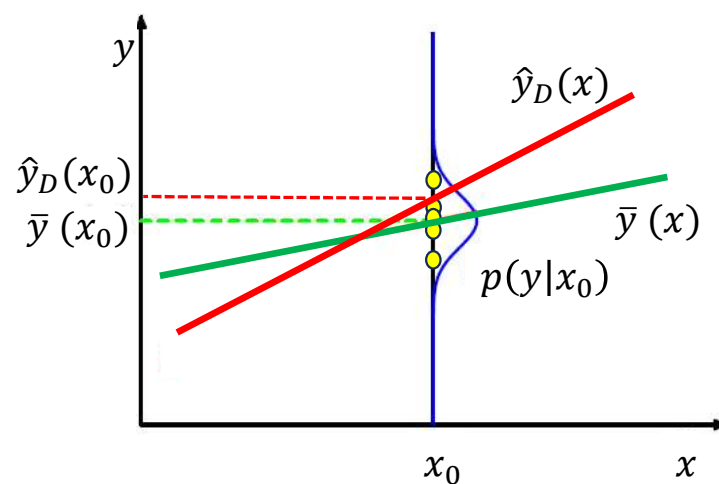
The Expected Test Error Given \hat{y}_D

- Given a learning algorithm \mathcal{A} , for a different training set \mathcal{D} , we will learn a different set of model parameters.
- In the context of multiple linear regression model, the weights we learn depends on the \mathcal{D} . Mathematically we have

$$\hat{y}_D \equiv \hat{y}_D(\mathbf{x}) = \mathbf{w}_D^T \mathbf{x}$$

- The expected test error given \hat{y}_D

$$E_{X,Y}[\mathcal{L}(\hat{y}_D, y)] = \int_{\mathbf{x}} \int_y \underbrace{\mathcal{L}(\hat{y}_D, y)}_{\substack{\uparrow \\ \text{loss function for a test point}}} p(\mathbf{x}, y) dy d\mathbf{x}$$



The Expected Test Error Given $\hat{y}_{\mathcal{D}}$

- If we use the squared loss, the **expected test error given $\hat{y}_{\mathcal{D}}$** is

$$E_{X,Y}[\mathcal{L}(\hat{y}_{\mathcal{D}}, y)] = E_{X,Y}[(\hat{y}_{\mathcal{D}}(\mathbf{x}) - y)^2] = \int_{\mathbf{x}} \int_y (\hat{y}_{\mathcal{D}}(\mathbf{x}) - y)^2 p(\mathbf{x}, y) dy dx$$

- Sample different training datasets \mathcal{D} I.I.D. from the same data distribution
- The **expected predictor** (i.e. regression function) given a model is

$$\bar{\hat{y}} \equiv \bar{\hat{y}}(\mathbf{x}) = E_{\mathcal{D}}[\hat{y}_{\mathcal{D}}] = \int_{\mathcal{D}} \hat{y}_{\mathcal{D}} p(\mathcal{D}) d\mathcal{D}$$

$\bar{\hat{y}}$ is a weighted average over functions

Note: No subscript \mathcal{D} here. \mathcal{D} is marginalized out

Expected Test Error given a Model

- For a fixed learning algorithm, random predictor $\hat{y}_{\mathcal{D}}$ and random target output y , we compute the **expected test error**

$$\begin{aligned} E_{\mathbf{X}, Y, \mathcal{D}}[\mathcal{L}(\hat{y}_{\mathcal{D}}, y)] &= E_{\mathbf{X}, Y, \mathcal{D}}[(\hat{y}_{\mathcal{D}}(\mathbf{x}) - y)^2] \\ &= \int_{\mathcal{D}} \int_{\mathbf{x}} \int_y (\hat{y}_{\mathcal{D}}(\mathbf{x}) - y)^2 p(\mathbf{x}, y) p(\mathcal{D}) dy dx d\mathcal{D} \end{aligned}$$

- The training set \mathcal{D} is independent from the test point (\mathbf{X}, Y) .
- We can also choose other forms of loss functions to measure the overall expected test error. Squared loss has a nice mathematical property to decompose the errors in regression problem.

Expected Test Error Decomposition

$$\begin{aligned}
 E_{X,Y,D}[(\hat{y}_D(\mathbf{x}) - y)^2] &= E_{X,Y,D} \left[\left((\hat{y}_D(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x})) + (\bar{\hat{y}}(\mathbf{x}) - y) \right)^2 \right] \\
 &= E_{X,\bar{Y},D}[(\hat{y}_D(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x}))^2] + E_{X,Y,\bar{D}}[(\bar{\hat{y}}(\mathbf{x}) - y)^2] \\
 &\quad + 2E_{X,Y,D}[(\hat{y}_D(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x}))(\bar{\hat{y}}(\mathbf{x}) - y)]
 \end{aligned}$$

- The last term is zero.

$$\begin{aligned}
 E_{X,Y,D}[(\hat{y}_D(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x}))(\bar{\hat{y}}(\mathbf{x}) - y)] &= E_{X,Y} \left[E_D[(\hat{y}_D(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x}))(\bar{\hat{y}}(\mathbf{x}) - y)] \right] \\
 &= E_{X,Y} \left[(\bar{\hat{y}}(\mathbf{x}) - y) E_D[(\hat{y}_D(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x}))] \right] = E_{X,Y}[(\bar{\hat{y}}(\mathbf{x}) - y)(E_D[\hat{y}_D(\mathbf{x})] - \bar{\hat{y}}(\mathbf{x}))] \\
 &= E_{X,Y}[(\bar{\hat{y}}(\mathbf{x}) - y)(\bar{\hat{y}}(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x}))] = 0
 \end{aligned}$$

Expected Test Error Decomposition

$$E_{X,Y,D}[(\hat{y}_D(\mathbf{x}) - y)^2] = \underbrace{E_{X,D}[(\hat{y}_D(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x}))^2]}_{\text{Variance}} + E_{X,Y}[(\bar{\hat{y}}(\mathbf{x}) - y)^2]$$

- The first term is the variance of the predictor
- The second term: we further decompose it

$$\begin{aligned} E_{X,Y}[(\bar{\hat{y}}(\mathbf{x}) - y)^2] &= E_{X,Y}[(\bar{\hat{y}}(\mathbf{x}) - \bar{y}(\mathbf{x})) + (\bar{y}(\mathbf{x}) - y)]^2 \\ &= \underbrace{E_{X,Y}[(\bar{\hat{y}}(\mathbf{x}) - \bar{y}(\mathbf{x}))^2]}_{\text{Bias}^2} + \underbrace{E_{X,Y}[(\bar{y}(\mathbf{x}) - y)^2]}_{\text{Noise}} + 2E_{X,Y}[(\bar{\hat{y}}(\mathbf{x}) - \bar{y}(\mathbf{x}))(\bar{y}(\mathbf{x}) - y)] \end{aligned}$$

Final Decomposition

$$E_{X,Y}[(\hat{y}(\mathbf{x}) - \bar{y}(\mathbf{x}))(\bar{y}(\mathbf{x}) - y)] = E_X[E_{Y|X}[(\bar{y}(\mathbf{x}) - y)](\hat{y}(\mathbf{x}) - \bar{y}(\mathbf{x}))]$$

$$= E_X[(\bar{y}(\mathbf{x}) - E_{Y|X}[y])(\hat{y}(\mathbf{x}) - \bar{y}(\mathbf{x}))]$$

• The third term is zero

$$= E_X[(\bar{y}(\mathbf{x}) - \bar{y}(\mathbf{x}))(\hat{y}(\mathbf{x}) - \bar{y}(\mathbf{x}))] = 0$$

-
- The expected test error is decomposed to

$$\underbrace{E_{X,Y,D}[(\hat{y}_D(\mathbf{x}) - y)^2]}_{\text{Expected Test Error}} = E_X \left[(\bar{y}(\mathbf{x}) - \bar{y}(\mathbf{x}))^2 \right] \quad \text{Bias}^2$$

$$+ E_{X,D}[(\hat{y}_D(\mathbf{x}) - \bar{y}(\mathbf{x}))^2] \quad \text{Variance}$$

$$+ E_{X,Y}[(\bar{y}(\mathbf{x}) - y)^2] \quad \text{Noise}$$

Summary of Decomposition

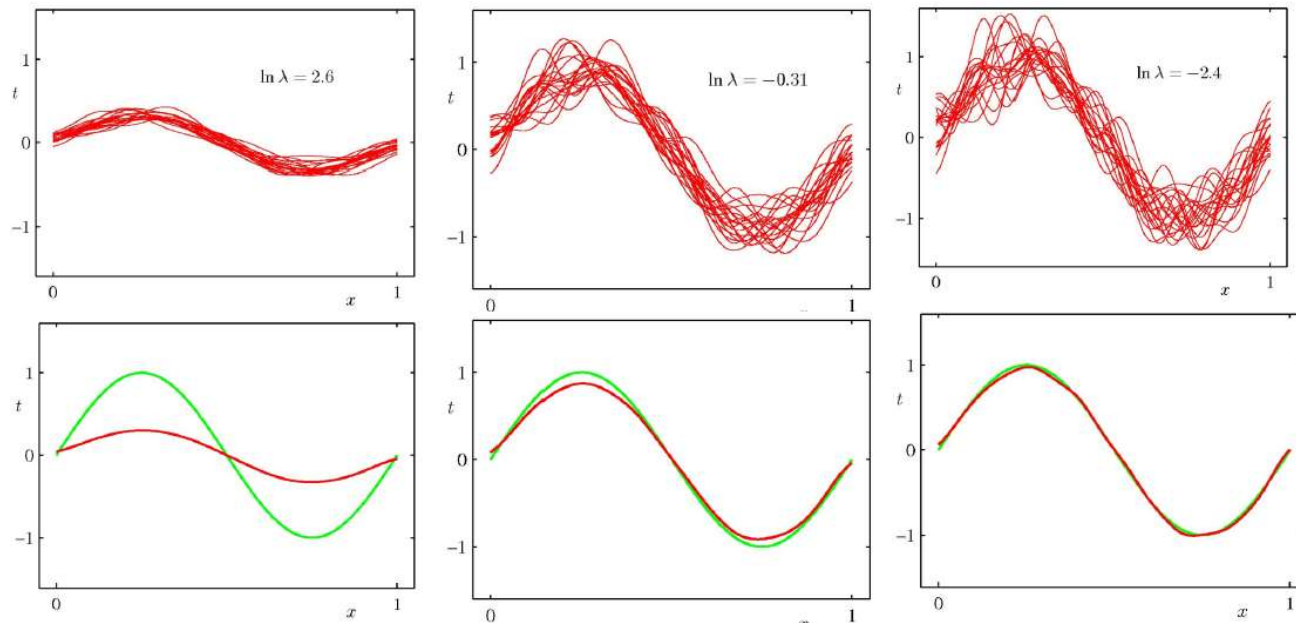
$$E_{X,Y,D}[(\hat{y}_D(\mathbf{x}) - y)^2] = \underbrace{E_X\left[\left(\bar{\hat{y}}(\mathbf{x}) - \bar{y}(\mathbf{x})\right)^2\right]}_{\text{Bias}^2} + \underbrace{E_{X,D}[(\hat{y}_D(\mathbf{x}) - \bar{\hat{y}}(\mathbf{x}))^2]}_{\text{Variance}} + \underbrace{E_{X,Y}[(\bar{y}(\mathbf{x}) - y)^2]}_{\text{Noise}}$$

Expected Test Error

The expected test errors come from three different sources

- **Bias:** High bias \approx underfitting
 - How **wrong** the **average prediction** is across different datasets
- **Variance:** High variance leads \approx overfitting
 - How much **predictions vary** when training datasets vary
- **Noise:** Bayes error
 - **Irreducible unpredictability** of the target

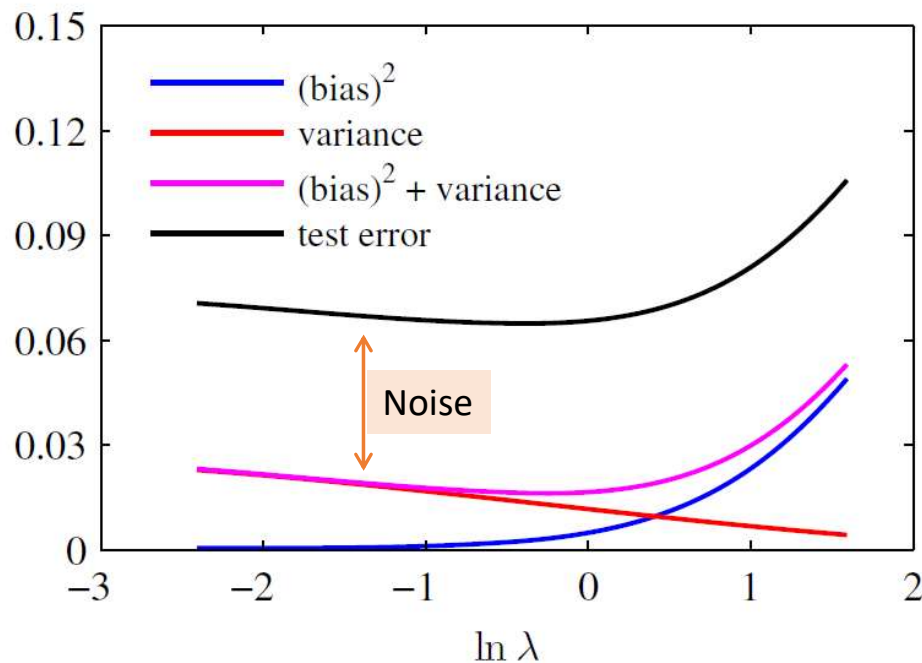
Basis Function Ridge Regression Example



[Image Courtesy Bishop's PRML]

Illustration of the dependence of bias and variance on model complexity, governed by a regularization parameter λ . There are 100 data sets, each having 25 data points, and there are 24 Gaussian basis functions in the model. The upper row shows the result of fitting the model to the data sets for various values of $\ln \lambda$ (only 20 out 100 fits shown). The lower row shows the corresponding average of the 100 fits (red) along the true value of the target function from which the data sets were generated (green)

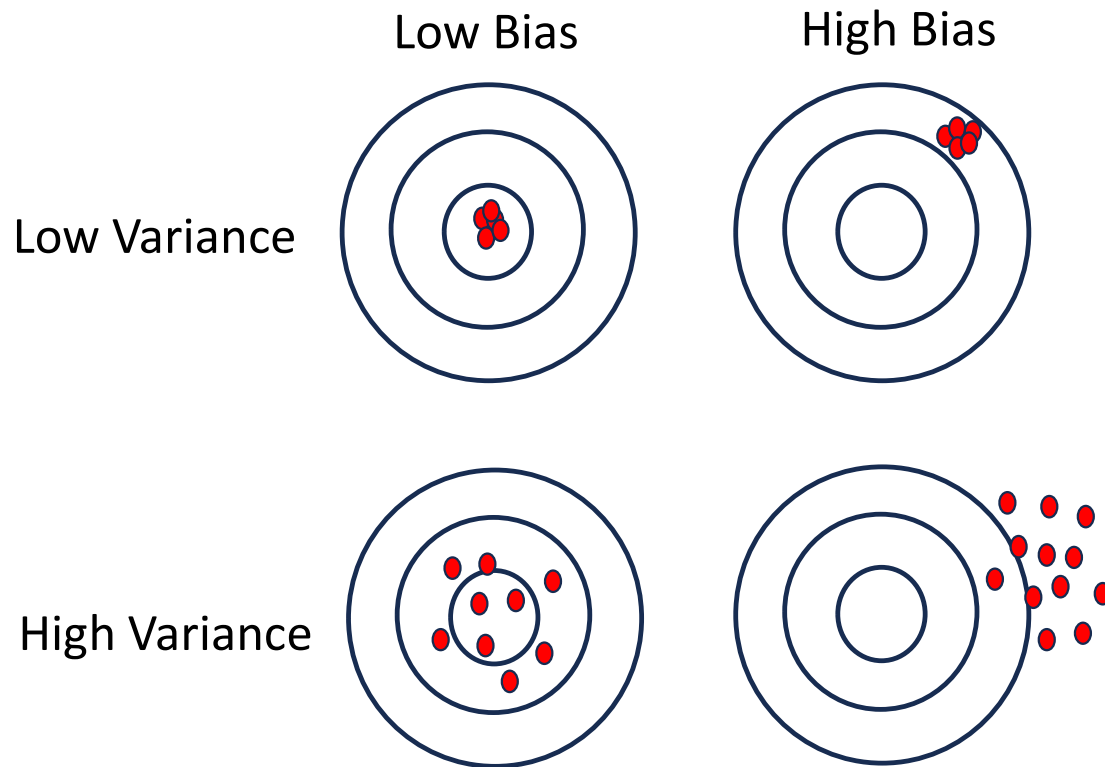
Bias Variance Plot



[Image Courtesy Bishop's PRML]

Plot of squared bias and variance, together with their sum, corresponding to the results shown in previous page. Also shown is the average test set error for a test data set size of 1000 points. The minimum value of $(\text{bias})^2 + \text{variance}$ occurs around $\ln \lambda = -0.31$, which is close to the value that gives the minimum error on the test data.

Bias and Variance Dart Board



- Throwing dart to make your prediction aiming the bull's eye
- Shaking Dart Board
 - Irreducible noise

Bayesian Methods

Bayesian Model Selection

Bayesian Model Selection

- What is a model?
 - In regression, should we use polynomials or RBFs as our basis function?
 - different degrees of polynomials, different number of RBFs
- Which model should we use?
- Cross validation
 - works well if we have enough data.
 - computationally expensive
- Bayesian methods use **posterior** probability distribution over models and select the one with higher posterior.

$$\begin{aligned}\mathcal{M}^* &= \operatorname{argmax}_{\mathcal{M}_i} P(\mathcal{M}_i|\mathcal{D}) \\ &= \operatorname{argmax}_{\mathcal{M}_i} P(D|\mathcal{M}_i)P(\mathcal{M}_i)\end{aligned}$$

$$P(\mathcal{M}_i|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M}_i)P(\mathcal{M}_i)}{P(\mathcal{D})}$$

Bayesian Model Selection Cont'd

- To compare two models \mathcal{M}_1 and \mathcal{M}_2 given data \mathcal{D} , consider

$$\frac{P(\mathcal{M}_1|\mathcal{D})}{P(\mathcal{M}_2|\mathcal{D})} = \underbrace{\frac{P(\mathcal{D}|\mathcal{M}_1)}{P(\mathcal{D}|\mathcal{M}_2)}}_{\text{Bayes Factor}} \frac{P(\mathcal{M}_1)}{P(\mathcal{M}_2)} \underset{?}{>} 1$$

- Without domain knowledge
 - We assume $P(\mathcal{M}_1) = P(\mathcal{M}_2)$
 - Select model based on $P(\mathcal{D}|\mathcal{M}_1)$, the **marginal data likelihood**
 - $P(\mathcal{D}|\mathcal{M}_i)$ incorporates parameters \mathbf{w}_1 for \mathcal{M}_1 and \mathbf{w}_2 for \mathcal{M}_2 through marginalization

$$\begin{aligned} p(\mathcal{D}|\mathcal{M}_i) &= \int p(\mathcal{D}, \mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i \\ &= \int p(\mathcal{D}|\mathbf{w}_i, \mathcal{M}_i) P(\mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i \end{aligned}$$

From Posterior to Marginal Likelihood

$$p(\mathcal{M}_i|\mathcal{D}) = \int p(\mathcal{M}_i, \mathbf{w}_i|\mathcal{D}) d\mathbf{w}_i = \int \frac{p(\mathcal{M}_i, \mathbf{w}_i, \mathcal{D})}{p(\mathcal{D})} d\mathbf{w}_i$$

$$= \int \frac{p(\mathcal{D}, \mathbf{w}_i|\mathcal{M}_i)p(\mathcal{M}_i)}{p(\mathcal{D})} d\mathbf{w}_i$$

$$= \int p(\mathcal{D}, \mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i \frac{p(\mathcal{M}_i)}{p(\mathcal{D})}$$

$$= \underbrace{\int p(\mathcal{D}|\mathbf{w}_i, \mathcal{M}_i)p(\mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i}_{p(\mathcal{D}|\mathcal{M}_i)} \frac{p(\mathcal{M}_i)}{p(\mathcal{D})}$$

Using Posterior over models to derive the marginal data likelihood integral form

Assume $p(\mathcal{M}_i) = p(\mathcal{M}_j)$, because $p(\mathcal{D})$ is constant, then only the integral term matters, which is the marginal data likelihood.

Digress: The Evidence Term

- Recall in parameter estimation, to find the MAP estimate of parameter \mathbf{w} , we compute

$$\max_{\mathbf{w}} p(\mathbf{w}|\mathcal{D}) = \max_{\mathbf{w}} \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

- Every term alone depends on a model
- Implicitly, we assumed a particular class of model

$$\max_{\mathbf{w}} p(\mathbf{w}|\mathcal{D}, \mathcal{M}) = \max_{\mathbf{w}} \frac{p(\mathcal{D}|\mathbf{w}, \mathcal{M})p(\mathbf{w}|\mathcal{M})}{p(\mathcal{D}|\mathcal{M})}$$

The Evidence Term Cont'd

- We ignored $p(\mathcal{D}|\mathcal{M})$ when estimating \mathbf{w} , since $p(\mathcal{D}|\mathcal{M})$ contains no \mathbf{w} .

$$\max_{\mathbf{w}} p(\mathbf{w}|\mathcal{D}, \mathcal{M}) = \max_{\mathbf{w}} \frac{p(\mathcal{D}|\mathbf{w}, \mathcal{M})p(\mathbf{w}|\mathcal{M})}{p(\mathcal{D}|\mathcal{M})}$$

- We are now interested in the posterior dependence on the model (i.e., not finding the most likely parameters), so $p(\mathcal{D}|\mathcal{M})$ matters

$$P(\mathcal{M}_i|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M}_i)P(\mathcal{M}_i)}{P(\mathcal{D})}$$

- $p(\mathcal{D}|\mathcal{M})$ is the evidence with which we select one model over the other. It tells how well the whole model explains the data, regardless of the specific parameters inside.

Remarks on Marginal Likelihood

- The model fits the best has an intuitive structure

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}_i, \mathcal{M}_i)p(\mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i$$

- More probable model will have both the likelihood and the prior large for the **same** model parameters (i.e. weight vectors)
- When parameters fit the data best happens to be the parameters that are highly likely under the prior probability distribution for the class of model, we have the best model.

Model Complexity

- Model complexity
 - can be measured by
 - the effective size of the parameter space
 - the entropy of the prior
 - discrete parameters: number of possible parameter values
 - continuous parameters: number of parameters and their ranges
- A model with a larger parameter space will assign lower prior probability to any one parameter value
 - A complex model spreads its prior probability mass more thinly over its parameter space since the prior must integrate to 1.

Bayesian Model Selection

- The marginal data likelihood

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}_i, \mathcal{M}_i)p(\mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i$$

- Given two models fit the data equally well for some range of parameters
- The more complex model will have a smaller prior
 - The probability mass is spread thinner
- The more complex model will have a lower marginal data likelihood
- Bayesian method provides a natural bias towards simpler models

Occam's Razor

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}_i, \mathcal{M}_i)p(\mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i$$

- **Principle of parsimony:** All things being equal, we prefer simpler explanation for phenomena
- In machine learning, we prefer simpler model under the assumption simpler models generalize better to future unseen data than more complex models.
- More complex models with more parameters are better able to fit noises in training data, leading to overfitting and producing poorer prediction.
- In the Bayesian model selection
 - $p(\mathbf{w}_i|\mathcal{M}_i)$, the prior over the model parameter space, plays a critical role
 - It controls how much our existing beliefs affect the model choice compared to how well the model fits the observed data (i.e. the likelihood $p(\mathcal{D}|\mathbf{w}_i, \mathcal{M}_i)$)

Example

- Suppose \mathcal{M}_1 is simple and \mathcal{M}_2 is complex
- \mathcal{M}_1 has one parameter: $w_1 \in \mathbb{R}$
- \mathcal{M}_2 has two parameters: $\mathbf{w}_2 \in \mathbb{R}^2$

$$w_1 \sim U[0,10]$$

$$\mathbf{w}_2 \sim U[0,10] \times [0,10]$$

$$p(w_1|\mathcal{M}_1) = \begin{cases} \frac{1}{10} & w_1 \in [0,10] \\ 0 & \text{otherwise} \end{cases}$$

$$p(\mathbf{w}_2|\mathcal{M}_2) = \begin{cases} \frac{1}{100} & \mathbf{w}_2 \in [0,10] \times [0,10] \\ 0 & \text{otherwise} \end{cases}$$

- The amount of uncertainty in \mathcal{M}_i is captured by the entropy of the prior. \mathcal{M}_2 has a higher entropy than \mathcal{M}_1

Example Cont'd

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}_i, \mathcal{M}_i)p(\mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i$$

- If \mathcal{M}_1 and \mathcal{M}_2 both fit the data similarly well. That is likelihoods are similar at MAP estimates of their parameters

$$p(\mathcal{D}|w_1^{MAP}, \mathcal{M}_1) \approx p(\mathcal{D}|\mathbf{w}_2^{MAP}, \mathcal{M}_2)$$

- In high dimension, the probability mass is spread to a much larger region, hence the prior over weights are smaller
- The simpler model where weights are in lower dimension has a bigger prior, hence a bigger marginal likelihood

MAP Estimate

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}_i, \mathcal{M}_i)p(\mathbf{w}_i|\mathcal{M}_i) d\mathbf{w}_i$$

- Consider $p(\mathbf{w}_i^{MAP})$ and $p(\mathbf{w}_i|\mathcal{M}_i)$, uniform PDFs
- The integral can be approximated by

$$p(\mathcal{D}|\mathcal{M}_i) \approx p(\mathcal{D}|\mathbf{w}_i^{MAP}, \mathcal{M}_i)p(\mathbf{w}_i^{MAP}|\mathcal{M}_i) \Delta\mathbf{w}_i^{posterior}$$

$$\approx p(\mathcal{D}|\mathbf{w}_i^{MAP}, \mathcal{M}_i) \frac{\Delta\mathbf{w}_i^{posterior}}{\Delta\mathbf{w}_i^{prior}}$$

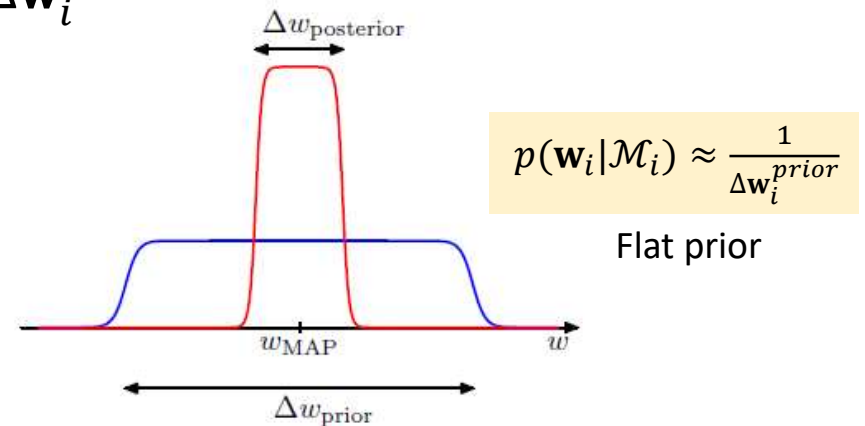


Figure 12.3: A visualization of the width-based evidence approximation. (Figure from *Pattern Recognition and Machine Learning* by Chris Bishop.)

MAP Estimate Cont'd $p(\mathcal{D}|\mathcal{M}_i) \approx p(\mathcal{D}|\mathbf{w}_i^{MAP}, \mathcal{M}_i) \frac{\Delta \mathbf{w}_i^{posterior}}{\Delta \mathbf{w}_i^{prior}}$

$$-\log p(\mathcal{D}|\mathcal{M}_i) = \underbrace{-\log p(\mathcal{D}|\mathbf{w}_i^{MAP}, \mathcal{M}_i)}_1 - \underbrace{\log \frac{\Delta \mathbf{w}_i^{posterior}}{\Delta \mathbf{w}_i^{prior}}}_2$$

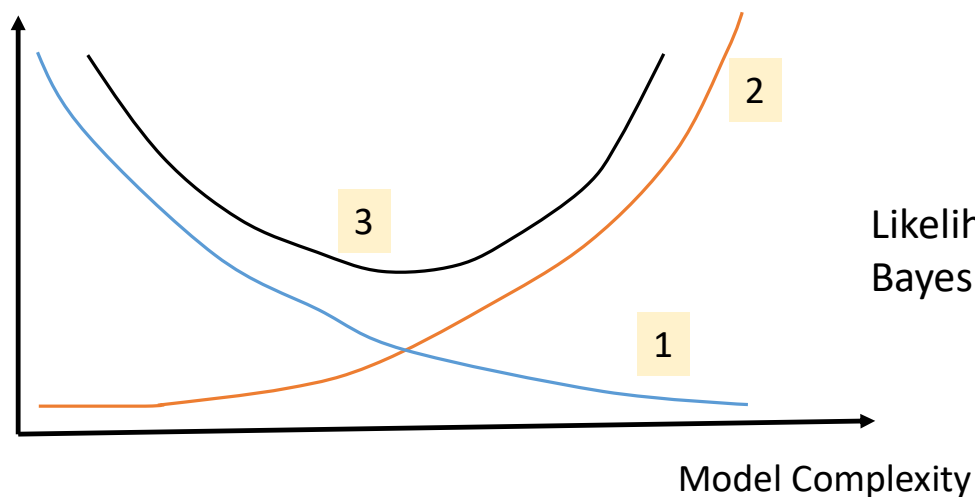
1 Good model fits the data well

2 When model is good, $\Delta \mathbf{w}_i^{posterior} = \Delta \mathbf{w}_i^{prior}$, the term is zero

- Usually $\Delta \mathbf{w}_i^{posterior} < \Delta \mathbf{w}_i^{prior}$, $-\log \frac{\Delta \mathbf{w}_i^{posterior}}{\Delta \mathbf{w}_i^{prior}}$ is minimal when $\Delta \mathbf{w}_i^{posterior} = \Delta \mathbf{w}_i^{prior}$
- More complex model has wider $\Delta \mathbf{w}_i^{prior}$, hence bigger $-\log \frac{\Delta \mathbf{w}_i^{posterior}}{\Delta \mathbf{w}_i^{prior}}$
- We can view term 2 as a penalty term on unnecessary complex models.

Balance btw Likelihood and Prior

$$\underbrace{-\log p(\mathcal{D}|\mathcal{M}_i)}_3 = \underbrace{-\log p(\mathcal{D}|\mathbf{w}_i^{MAP}, \mathcal{M}_i)}_1 - \underbrace{\log \frac{\Delta \mathbf{w}_i^{posterior}}{\Delta \mathbf{w}_i^{prior}}}_2$$



Likelihood and Prior are two competing factors
Bayesian model incorporates both factors

Acknowledgement

- Prof. David Fleet developed the course. He made his notes and courseware available to all of us.
- Prof. Francisco (Paco) Estrada shared his assignments and insights.
- Prof. Rawad A. Assi shared past assignments and advices.