

## Lecture 7. Regularization and model selection

Honglak Lee

2/3/2025



## Outline

- MLE, MAP
  - Maximum Likelihood
  - MAP
- Bias-Variance Tradeoff
- Model selection
  - Cross validation

### MLE vs. MAP

- Maximum Likelihood Estimation (MLE)
  - Objective: Log-likelihood  
 $\log P(\mathbf{D}|\mathbf{w})$
  - Example: linear regression (w/o regularization)
- Maximum a Posteriori (MAP)
  - Objective: Log-likelihood + Log-Prior  
 $\log P(\mathbf{D}|\mathbf{w}) + \log P(\mathbf{w})$
  - Example: regularized linear regression

### Maximum Likelihood

- Objective function (to maximize): log-likelihood
  - discriminative model:
$$\log P(\mathbf{D}|\mathbf{w}) = \log \prod_{n=1}^N P(y^{(n)}|\phi(\mathbf{x}^{(n)}), \mathbf{w})$$

IID: Independent and Identically Distributed

$$= \sum_{n=1}^N \log P(y^{(n)}|\phi(\mathbf{x}^{(n)}), \mathbf{w})$$

- generative model:

$$\log P(\mathbf{D}|\mathbf{w}) = \sum_{n=1}^N \log P(y^{(n)}, \mathbf{x}^{(n)}|\mathbf{w})$$

Note: Today, we will look into the discriminative setting, but the theory/analysis will similarly apply to generative models too.

### Maximum Likelihood

- Objective function (to maximize): log-likelihood
 
$$\log P(\mathbf{D}|\mathbf{w}) = \log \prod_{n=1}^N P(y^{(n)}|\phi(\mathbf{x}^{(n)}), \mathbf{w})$$

IID: Independent and Identically Distributed

- Linear regression (without regularization)

- Logistic regression (without regularization)

- Problem of Maximum Likelihood: risk of overfitting

- when the model is too complex for the number of training samples

### Maximum a Posteriori (MAP)

$$P(\mathbf{w}|\mathbf{D}) = \frac{P(\mathbf{D}|\mathbf{w})P(\mathbf{w})}{P(\mathbf{D})}$$

$$\propto P(\mathbf{D}|\mathbf{w})P(\mathbf{w})$$

- Assumes prior distribution  $P(\mathbf{w})$

- Point estimate using Bayes rule:

$$\arg \max_{\mathbf{w}} P(\mathbf{w}|\mathbf{D}) = \arg \max_{\mathbf{w}} P(\mathbf{D}|\mathbf{w})P(\mathbf{w})$$

- Objective function to maximize:

$$\begin{aligned} \log P(\mathbf{D}|\mathbf{w})P(\mathbf{w}) &= \log \prod_{n=1}^N P(y^{(n)}|\phi(\mathbf{x}^{(n)}), \mathbf{w}) + \log P(\mathbf{w}) \\ &= \sum_{n=1}^N \log P(y^{(n)}|\phi(\mathbf{x}^{(n)}), \mathbf{w}) + \log P(\mathbf{w}) \end{aligned}$$

Note: this objective is for discriminative models, but the formulation is similar for generative models.

### MAP

- Isotropic Gaussian (e.g. L2 norm) is a popular prior (regularizer):

$$P(\mathbf{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I})$$

$$\log P(\mathbf{w}) = -\frac{\lambda}{2}\|\mathbf{w}\|^2 + \text{const}$$

- Example:

- L2-regularized linear regression
- L2-regularized logistic regression

### Check: Gaussian Prior and L2 regularization

- Gaussian prior distribution for  $\mathbf{w}$

$$P(\mathbf{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I})$$

$$= \text{const} * \exp\left(-\frac{1}{2}\mathbf{w}^\top(\lambda^{-1}\mathbf{I})^{-1}\mathbf{w}\right)$$

$$= \text{const} * \exp\left(-\frac{\lambda}{2}\mathbf{w}^\top\mathbf{w}\right)$$

- Taking log

$$\log P(\mathbf{w}) = \text{const} - \frac{\lambda}{2}\mathbf{w}^\top\mathbf{w} = \text{const} - \frac{\lambda}{2}\|\mathbf{w}\|^2$$

"Gaussian Prior" for  $\mathbf{w}$  and L2 regularization are equivalent.

## Putting together: MAP

$$P(\mathbf{w}|\mathbf{D}) = \frac{P(\mathbf{D}|\mathbf{w})P(\mathbf{w})}{P(\mathbf{D})} \propto P(\mathbf{D}|\mathbf{w})P(\mathbf{w})$$

- Maximizing posterior of  $\mathbf{w}$  given  $\mathbf{D}$  w/ Bayes rule:

$$\arg \max_{\mathbf{w}} P(\mathbf{w}|\mathbf{D}) = \arg \max_{\mathbf{w}} P(\mathbf{D}|\mathbf{w})P(\mathbf{w})$$

- Objective function to maximize:

$$\log P(\mathbf{D}|\mathbf{w})P(\mathbf{w}) = \sum_{n=1}^N \log P(y^{(n)}|\phi(\mathbf{x}^{(n)}), \mathbf{w}) + \log P(\mathbf{w})$$

$$- \frac{\lambda}{2} \|\mathbf{w}\|^2$$

assuming Gaussian prior for  $\mathbf{w}$

- Example:

- L2-regularized linear regression
- L2-regularized logistic regression

9

## MAP for Regularized Least Squares

$$\text{From } P(y^{(n)}|\phi(\mathbf{x}^{(n)}), \mathbf{w}, \beta) = \mathcal{N}(y^{(n)}|\mathbf{w}^\top \phi(\mathbf{x}^{(n)}), \beta^{-1}) \\ = \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2\right)$$

$$\begin{aligned} \text{Derive: } \log P(\mathbf{D}|\mathbf{w})P(\mathbf{w}) &= \sum_{n=1}^N \log P(y^{(n)}|\phi(\mathbf{x}^{(n)}), \mathbf{w}) + \log P(\mathbf{w}) \\ &= \sum_{n=1}^N \log \left( \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2\right) \right) + \text{const} - \frac{\lambda}{2} \|\mathbf{w}\|^2 \\ &= \frac{N}{2} \log \beta - \frac{N}{2} \log 2\pi - \sum_{n=1}^N \frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2 + \text{const} - \frac{\lambda}{2} \|\mathbf{w}\|^2 \end{aligned}$$

Minimize:

$$\tilde{E}(\mathbf{w}) = \sum_{n=1}^N \frac{\beta}{2} \|y^{(n)} - \mathbf{w}^\top \phi(\mathbf{x}^{(n)})\|^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 + \text{const wrt } \mathbf{w}$$

10

## Solving Regularized Least Squares

- Consider the error function:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

- With the sum-of-squares error function and a quadratic regularizer, we get Penalize large coefficient values

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

Remark:  $\beta = 1$  for simplicity

$\lambda$  is called the regularization coefficient.

- which is minimized by  $\mathbf{w}_{\text{ML}} = (\lambda \mathbf{I} + \Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}$

11

## Derivation

### Objective function

$$\begin{aligned} \tilde{E}(\mathbf{w}) &= \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 \\ &= \frac{1}{2} \mathbf{w}^\top \Phi^\top \Phi \mathbf{w} - \mathbf{w}^\top \Phi^\top \mathbf{y} + \frac{1}{2} \mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \end{aligned}$$

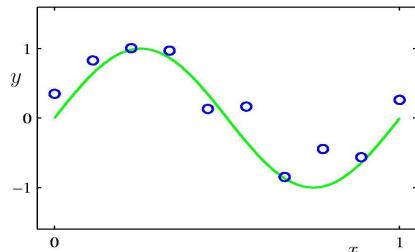
Compute gradient and set it zero:

$$\begin{aligned} \nabla_{\mathbf{w}} \tilde{E}(\mathbf{w}) &= \nabla_{\mathbf{w}} \left[ \frac{1}{2} \mathbf{w}^\top \Phi^\top \Phi \mathbf{w} - \mathbf{w}^\top \Phi^\top \mathbf{y} + \frac{1}{2} \mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \right] \\ &= \Phi^\top \Phi \mathbf{w} - \Phi^\top \mathbf{y} + \lambda \mathbf{w} \\ &= (\lambda \mathbf{I} + \Phi^\top \Phi) \mathbf{w} - \Phi^\top \mathbf{y} \\ &= 0 \end{aligned}$$

Therefore, we get:  $\mathbf{w}_{\text{ML}} = (\lambda \mathbf{I} + \Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y}$

12

## Revisiting Polynomial Curve Fitting

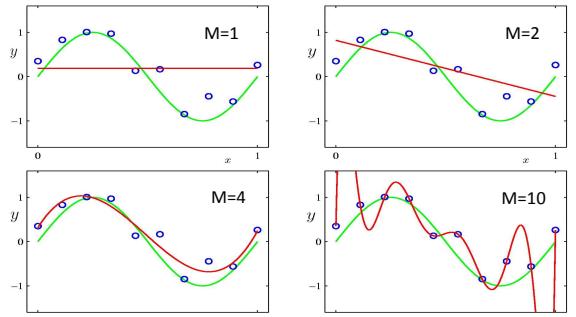


$$h(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_{M-1} x^{M-1} = \sum_{j=0}^{M-1} w_j x^j$$

## Maximum Likelihood (in Linear Regression)

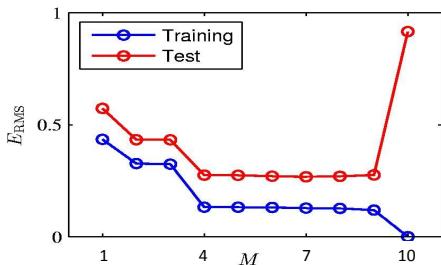
- Choosing the right set of features is important

– Watch out for underfitting/overfitting



14

## Underfitting vs. overfitting



Root-Mean-Square (RMS) Error:  $E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$

Note: RMS is just an evaluation metric, not the training objective

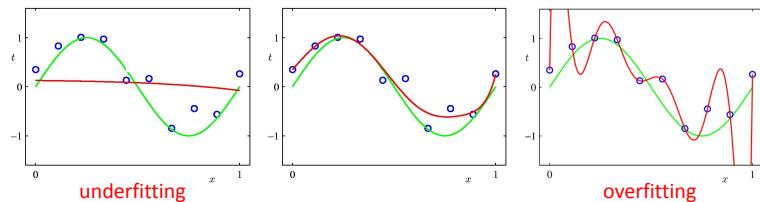
## L2 Regularization with different $\lambda$ values

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^\top \phi(\mathbf{x}^{(n)}) - y^{(n)})^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

$\lambda = 10$

$\lambda = 0.001$

$\lambda = 0$

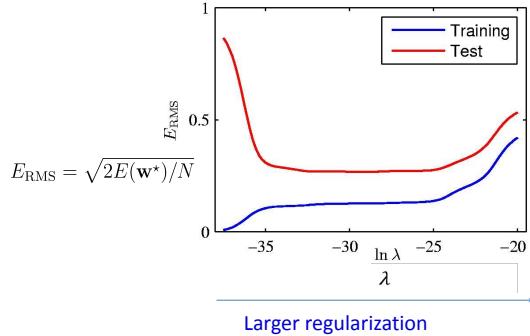


Takeway: Picking the right regularization coefficient  $\lambda$  is important

15

16

## L2 Regularization: $E_{\text{RMS}}$ vs. $\lambda$



NOTE: For simplicity of presentation, we divided the data into training set and test set. However, it's **not** legitimate to find the optimal hyperparameter based on the test set. We will talk about legitimate ways of doing this when we cover model selection and cross-validation.

17

## Polynomial Coefficients

	$\lambda=0$	$\lambda=0.001$	$\lambda=10$
$w_0^*$	0.35	0.35	0.13
$w_1^*$	232.37	4.74	-0.05
$w_2^*$	-5321.83	-0.77	-0.06
$w_3^*$	48568.31	-31.97	-0.05
$w_4^*$	-231639.30	-3.89	-0.03
$w_5^*$	640042.26	55.28	-0.02
$w_6^*$	-1061800.52	41.32	-0.01
$w_7^*$	1042400.18	-45.95	-0.00
$w_8^*$	-557682.99	-91.53	0.00
$w_9^*$	125201.43	72.68	0.01

## Summary: L2 regularized linear regression

- Simple modification of linear regression
- L2 regularization controls the tradeoff between “fitting error” and “complexity”.
  - Small L2 regularization results in complex models (but with risk of overfitting)
  - Large L2 regularization results in simple models (but with risk of underfitting).
- It is important to find an optimal regularization that balances between the two.

17

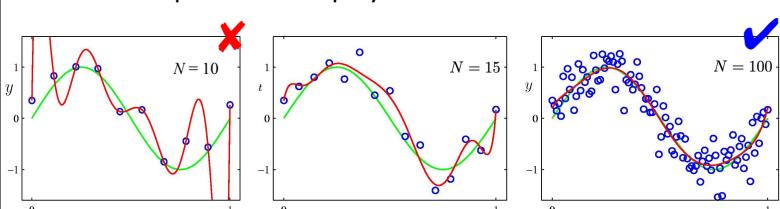
## How can we avoid overfitting?

- More training data
  - Always helps
- Choosing the right set of features of the model
- Regularization (e.g. MAP)
  - Penalize too complex models

18

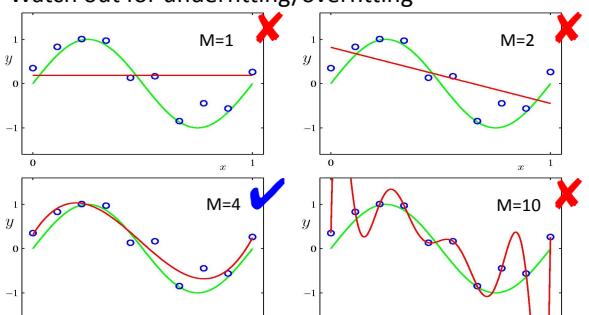
## Recap: More training data

- The more complicated the model, the more it benefits from more data (by avoiding overfitting)
  - Example: 9<sup>th</sup> order polynomial



## Recap: Choosing the right features

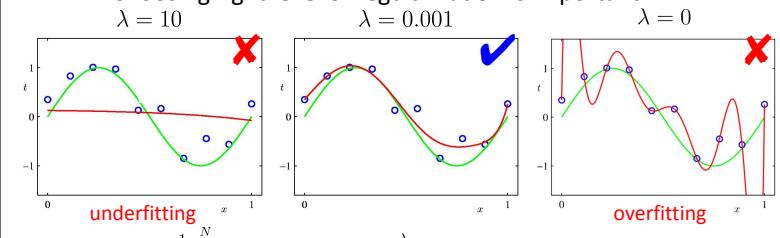
- Choosing the right set of features is important
  - Watch out for underfitting/overfitting



22

## Recap: Regularization

- Regularization can implicitly control the complexity of models
  - Example: 9<sup>th</sup> order polynomial
  - Choosing right level of regularization is important

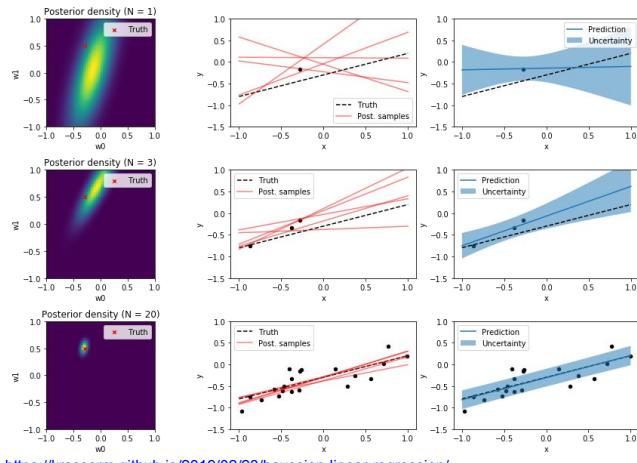


## Remarks: Fully Bayesian Approach

- Assumes prior distribution:  $P(\mathbf{w})$
- Goal: Estimate the posterior  $P(\mathbf{w}|D)$  from D
 
$$P(\mathbf{w}|D) \propto P(D|\mathbf{w})P(\mathbf{w})$$
  - Goal is not a point estimate!! (Unlike ML or MAP)
- Usually we want an analytic form of the posterior
  - Often, conjugate priors are used
- Example:
  - Bayesian linear regression
  - Gaussian Process

23

24



<https://krasserm.github.io/2019/02/23/bayesian-linear-regression/>

25

## ML vs. MAP vs. Fully Bayesian (summary)

- Maximum Likelihood
  - Objective: Log-likelihood:  $\log P(D|w)$
  - Example: linear regression (w/o regularization)
  - Note: log-likelihood for discriminative vs generative models
    - Discriminative models: Conditional log-likelihood  
 $\sum_i \log P(y^{(i)} | \mathbf{x}^{(i)}, \mathbf{w})$
    - Generative models: Joint log-likelihood  
 $\sum_i \log P(y^{(i)}, \mathbf{x}^{(i)} | \mathbf{w})$
- MAP (Maximum a Posteriori)
  - Objective: Log-likelihood + Log-Prior  
 $\log P(D|w) + \log P(w)$
  - Example: Regularized linear regression
- (Fully) Bayesian:
  - Try to fully estimate the posterior  $P(w | x, y)$ , and sample w from this.
  - Example: Bayesian linear regression (see Bishop book)

26

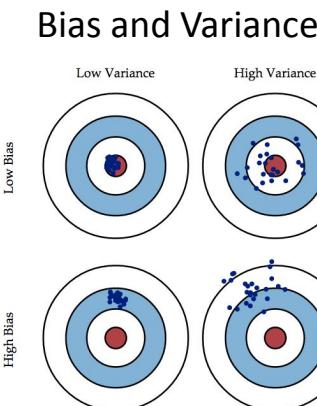
## Bias-Variance Tradeoff

## The Bias-Variance Decomposition

- Setting:
  - Given a distribution of  $P(\mathbf{x}, y)$
  - Sample training data  
 $D_{\text{train}} = \{(\mathbf{x}^{(n)}, y^{(n)}) : n = 1, \dots, N\} \sim P(\mathbf{x}, y)$
  - Train a learning algorithm on D
- Depending on samples, learning algorithm can still give different results (MLE, MAP, etc.)
- Goal: We want to learn a model with
  - Small bias (i.e. how well a model fits the data on average?)
  - Small variance (i.e. how stable a model is wrt data samples?)

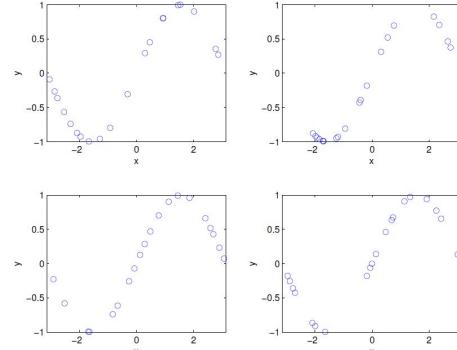
27

28



## The Bias-Variance Decomposition

- Example: samples from the sinusoidal function  $y=\sin(x)$



29

30

## The Bias-Variance Decomposition (advanced)

- Expected squared loss:

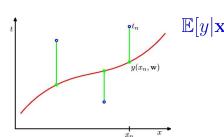
$$\mathbb{E}[L] = \int \int \{h(\mathbf{x}) - y\}^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

$$\mathbb{E}[L] = \boxed{\int \{h(\mathbf{x}) - \mathbb{E}[y|\mathbf{x}]\}^2 p(\mathbf{x}) dx} + \int \int \{\mathbb{E}[y|\mathbf{x}] - y\}^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

– where  $\mathbb{E}[y|\mathbf{x}] = \int y p(y|\mathbf{x}) dy$

- The second term corresponds to the noise inherent in the random variable y.

- What about the first term?



## The Bias-Variance Decomposition

- Suppose we were given multiple data sets, each of size N. Any particular data set (sampled from a distribution), D, will give a particular function  $h(\mathbf{x}; D)$ . We then have

$$\mathbb{E}_D \int \{h(\mathbf{x}; D) - \mathbb{E}[y|\mathbf{x}]\}^2 p(\mathbf{x}) dx$$

$$= \underbrace{\int \{\mathbb{E}_D[h(\mathbf{x}; D)] - \mathbb{E}[y|\mathbf{x}]\}^2 p(\mathbf{x}) dx}_{(\text{bias})^2} + \underbrace{\int \mathbb{E}_D [\{h(\mathbf{x}; D) - \mathbb{E}_D[h(\mathbf{x}; D)]\}^2] p(\mathbf{x}) dx}_{\text{variance}}$$

31

32

## The Bias-Variance Decomposition

- Expected loss

$$\mathbb{E}[L] = \int \int \{h(\mathbf{x}) - y\}^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

expected loss = (bias)<sup>2</sup> + variance + noise

- where

$$\mathbb{E}[y|\mathbf{x}] = \int y p(y|\mathbf{x}) dy$$

$$(\text{bias})^2 =$$

$$\text{variance} =$$

$$\text{noise} =$$

33

## The Bias-Variance Decomposition

- Expected loss

$$\mathbb{E}[L] = \int \int \{h(\mathbf{x}) - y\}^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

expected loss = (bias)<sup>2</sup> + variance + noise

- where

$$\mathbb{E}[y|\mathbf{x}] = \int y p(y|\mathbf{x}) dy$$

$$(\text{bias})^2 = \int \{\mathbb{E}_D[h(\mathbf{x}; D)] - \mathbb{E}[y|\mathbf{x}]\}^2 p(\mathbf{x}) d\mathbf{x}$$

$$\text{variance} = \int \mathbb{E}_D [\{h(\mathbf{x}; D) - \mathbb{E}_D[h(\mathbf{x}; D)]\}^2] p(\mathbf{x}) d\mathbf{x}$$

$$\text{noise} = \int \int \{\mathbb{E}[y|\mathbf{x}] - y\}^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

34

## Proof: Details

D: training data; x: test example; y: label of x

$$\begin{aligned} \mathbb{E}[L] &= \mathbb{E}_{x,y,D} [(h(\mathbf{x}; D) - y)^2] \\ &= \mathbb{E}_{x,y,D} [(h(\mathbf{x}; D) - \mathbb{E}[y|\mathbf{x}])^2] + \mathbb{E}_{x,y,D} [(y - \mathbb{E}[y|\mathbf{x}])^2] \\ &= \mathbb{E}_{x,D} [(h(\mathbf{x}; D) - \mathbb{E}[y|\mathbf{x}])^2] + \text{const} \quad \text{because this second term doesn't depend on the model} \\ &= \mathbb{E}_x [\mathbb{E}_D [(h(\mathbf{x}; D) - \mathbb{E}[y|\mathbf{x}])^2]] + \text{const} \\ &= \mathbb{E}_x [\mathbb{E}_D [(h(\mathbf{x}; D) - \mathbb{E}_D[h(\mathbf{x}; D)] + \mathbb{E}_D[h(\mathbf{x}; D)] - \mathbb{E}[y|\mathbf{x}])]^2]] + \text{const} \\ &= \mathbb{E}_x [\mathbb{E}_D [\{h(\mathbf{x}; D) - \mathbb{E}_D[h(\mathbf{x}; D)]\}^2]] \\ &\quad + 2\mathbb{E}_x [\mathbb{E}_D [\{h(\mathbf{x}; D) - \mathbb{E}_D[h(\mathbf{x}; D)]\} \{ \mathbb{E}_D [h(\mathbf{x}; D)] - \mathbb{E}[y|\mathbf{x}] \}]] \\ &\quad + \mathbb{E}_x [\{\mathbb{E}_D [h(\mathbf{x}; D)] - \mathbb{E}[y|\mathbf{x}]\}^2] + \text{const} \\ &= \mathbb{E}_x [\mathbb{E}_D [\{h(\mathbf{x}; D) - \mathbb{E}_D[h(\mathbf{x}; D)]\}^2]] + \mathbb{E}_x [\{\mathbb{E}_D [h(\mathbf{x}; D)] - \mathbb{E}[y|\mathbf{x}]\}^2] + \text{const} \end{aligned}$$

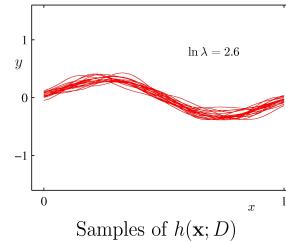
Variance

Bias

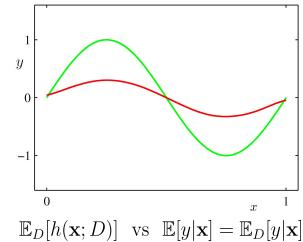
35

## Example: regularized linear regression

- Example: 100 data sets from the sinusoidal, varying the degree of regularization,  $\lambda$ .



Samples of  $h(\mathbf{x}; D)$



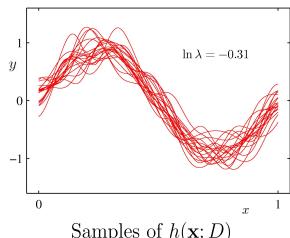
$\mathbb{E}_D[h(\mathbf{x}; D)]$  vs  $\mathbb{E}[y|\mathbf{x}] = \mathbb{E}_D[y|\mathbf{x}]$

Bishop book p.150

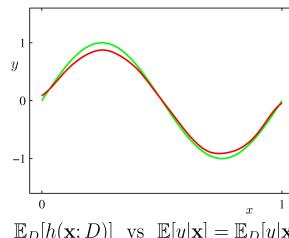
Setting : samples of 100 data sets (each curve is regression model for each sampled data set)  
N = 25 data points, 24 Gaussian basis functions (M = 25 including the bias parameter)

## Example: regularized linear regression

- Example: 100 data sets from the sinusoidal, varying the degree of regularization,  $\lambda$ .



Samples of  $h(\mathbf{x}; D)$



$\mathbb{E}_D[h(\mathbf{x}; D)]$  vs  $\mathbb{E}[y|\mathbf{x}] = \mathbb{E}_D[y|\mathbf{x}]$

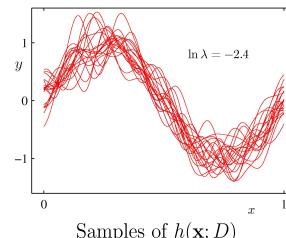
Bishop book p.150

Setting : samples of 100 data sets (each curve is regression model for each sampled data set)  
N = 25 data points, 24 Gaussian basis functions (M = 25 including the bias parameter)

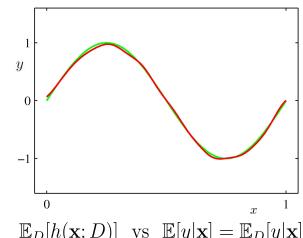
37

## Example: regularized linear regression

- Example: 100 data sets from the sinusoidal, varying the degree of regularization,  $\lambda$ .



Samples of  $h(\mathbf{x}; D)$



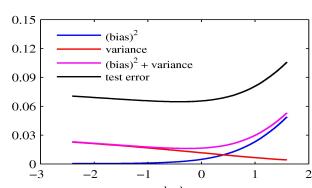
$\mathbb{E}_D[h(\mathbf{x}; D)]$  vs  $\mathbb{E}[y|\mathbf{x}] = \mathbb{E}_D[y|\mathbf{x}]$

Bishop book p.150

38

## The Bias-Variance Trade-off

- An over-regularized model (large  $\lambda$ ) will have a high bias and low variance.
- An under-regularized model (small  $\lambda$ ) will have a high variance and low bias.
- It is important to find a good balance between the two.
  - Typically done by cross validation (will be covered later)



39

## Model Selection

40

## Choosing the right model

- For polynomial curve fitting (least squares), which value of M should we choose?
- For regularized least squares / logistic regression, which  $\lambda$  value should we choose?
- Generally, given a set of models  $M = \{M_1, M_2, \dots, M_d\}$ , how can we choose optimal  $M_*$ ?
  - Model:
    - Class (or set) of hypotheses: learning algorithm, hyperparameters, etc.
    - Fixed during training
  - Parameters:
    - Aka hypothesis (e.g.  $w$  for logistic regression/linear regression)
    - Can be trained based on data

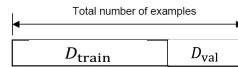
41

## Simple idea that doesn't work

- Given data D, train each model  $M_i$  on D, to get a hypothesis  $h_i$  (for model i)
- Pick the hypothesis with the smallest training error
  - Problematic: this leads to overfitting

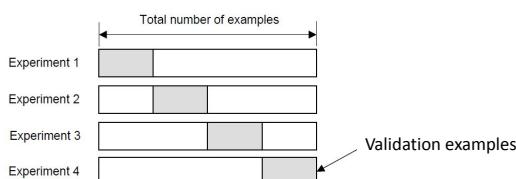
42

## Cross validation

- Hold out cross validation
  - Randomly split D into  $D_{train}$  and  $D_{val}$ 
    - $D_{train}$  should have more than  $D_{val}$  – something like 70%:30% is a typical split
    - $D_{val}$  also known as the hold-out validation set
  - Train each model  $M_i$  on  $D_{train}$  only to get some hypothesis  $h_i$
  - Select and output hypothesis  $h_i$  that had the smallest error on the hold out validation set
- Disadvantage:
  - Wastes the data in  $D_{val}$  (can't train on it)
  - Some data only used only for training, some only for validation

43

## K-fold Cross validation

- Create a K-fold partition of the dataset
  - For each of K experiments, use K-1 folds for training and the remaining one for validating
- The true error is estimated as the average error rate

44

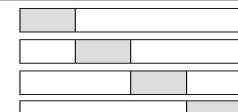
## K fold cross validation

- 1. Randomly split D into K disjoint subsets of  $N/K$  training samples each
  - Subsets are named  $D_1, \dots, D_K$
- 2. For each model  $M_i$ , we evaluate as follows:
  - For each fold ( $k = 1, \dots, K$ ):
    - Train  $M_i$  on  $D_1 \cup \dots \cup D_{k-1} \cup D_{k+1} \cup \dots \cup D_K$  (i.e. train on all data except  $D_k$ ) to get some hypothesis  $h_i(k)$
    - Test the hypothesis  $h_i(k)$  on  $D_k$  to get error/loss  $L_i(k)$
    - Estimated generalization error of  $M_i$  is average of all errors:
$$\hat{\epsilon}_i = \frac{1}{K} \sum_k \epsilon_i(k)$$
- 3. Pick  $M_*$  with the lowest estimated generalization error  $\hat{\epsilon}_i$  and retrain  $M_*$  on the entire training set S.
  - Resulting hypothesis is our final answer

45

## K-fold Cross validation

- Special case:  $K=N$ 
  - Called, leave-one-out cross validation (LOO CV)
  - Expensive, but wastes least amount of training data for cross validation. Could be useful when the data size is not big.
  - Computationally unsuitable when you have a large data
- Which K value should we use?
  - Popular choice of  $K = 10, 5$
  - For large data, then  $K = 3$  could be enough.
  - For small amount of data, you may need LOOCV to utilize as many training examples as possible.



46

## Three way data splits

- If model selection and true error estimates are to be computed simultaneously, the data needs to be divided into three disjoint sets
- Training set:** a set of examples used for learning: to fit the parameters of the classifier
  - Used for training parameters ( $w$  in logistic regression) given a fixed hyperparameters
- Validation set:** a set of examples used to tune the hyperparameters of a classifier
  - we would use the validation set to find the “optimal” hyperparameters
  - E.g., L2 regularization parameter for L2 logistic regression
- Test set:** a set of examples used only to assess the performance of a fully-trained classifier
  - After assessing the final model with the test set, **YOU MUST NOT** further tune the model
  - i.e., test set must be used only for evaluation, NOT for “tuning” your models & hyperparameters.

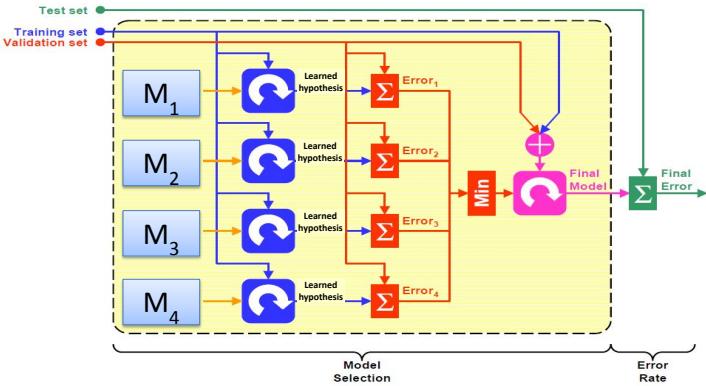
47

## Practical Recipe for Training, Model Selection, Evaluation (all put together)

- Divide the available data into **training**, **validation** and **test set**
- Repeat the following steps with different models (and hyperparameters)
  - Select a model (and hyperparameters)
  - Train the model using the **training set**
  - Evaluate the model using the **validation set**
- Select the best model (and hyperparameter)
- (optional) Re-train the best model above using data from the **training set** and **validation set** combined
- Assess this final model using the **test set**

48

## Procedure Illustration



49

## What's coming up

- Beginning next lecture and up through next week, we will talk about kernel methods

Any feedback (about lecture, slide, homework, project, etc.)?

(via anonymous google form: <https://forms.gle/fpYmiBtG9Me5gbP37>)



Change Log of lecture slides:

<https://docs.google.com/document/d/e/2PACX-1vSSIHjiklynK7rKFsr1-5GYxvBCEW8UPtpSfCR9AR6M1l7K9ZQEmxfFwaWaW7kLDxusthsF8WICyZI-/pub>

51

50