The Basic of Statistical Learning

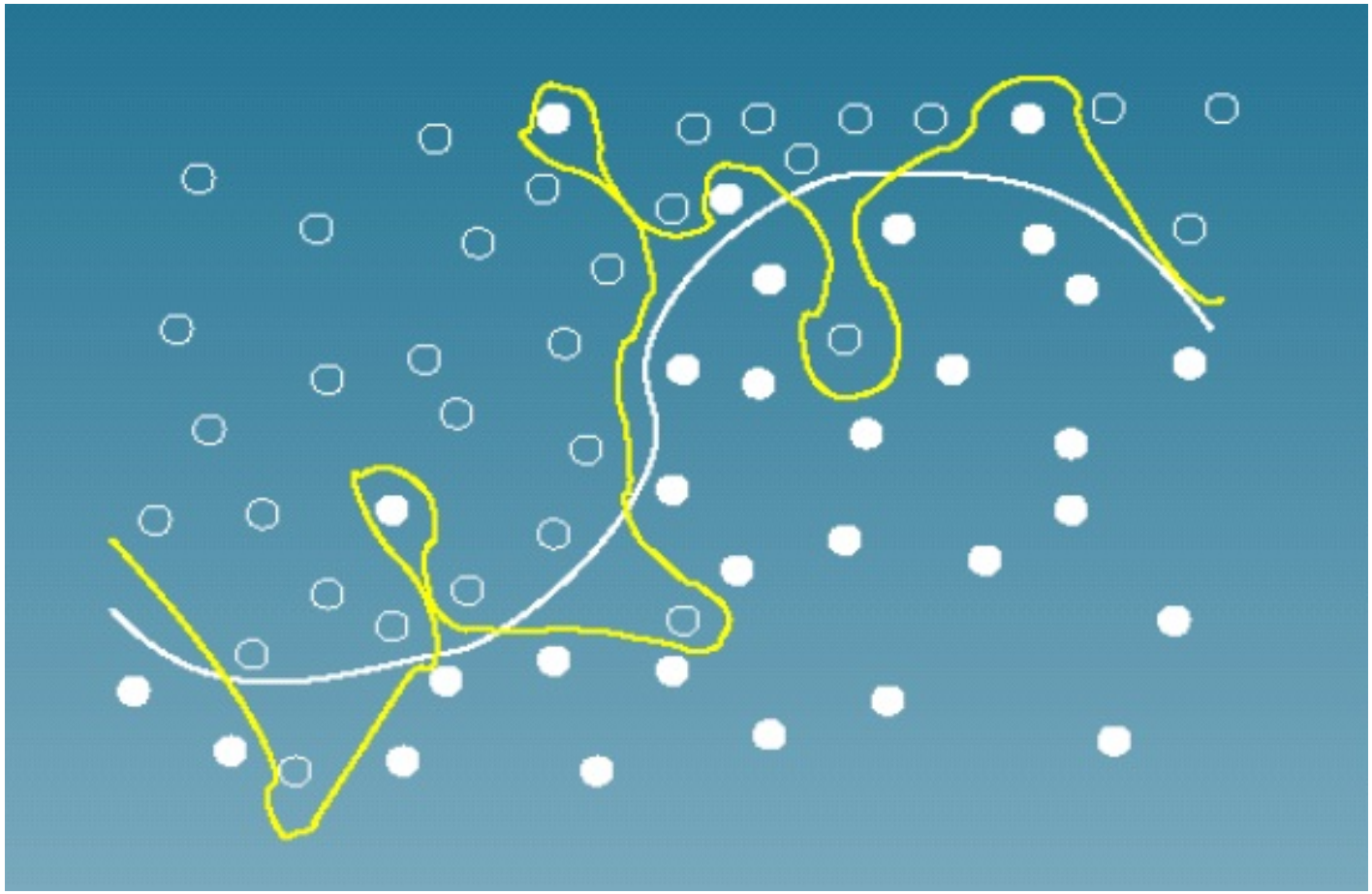
Hong Chang

Institute of Computing Technology,
Chinese Academy of Sciences

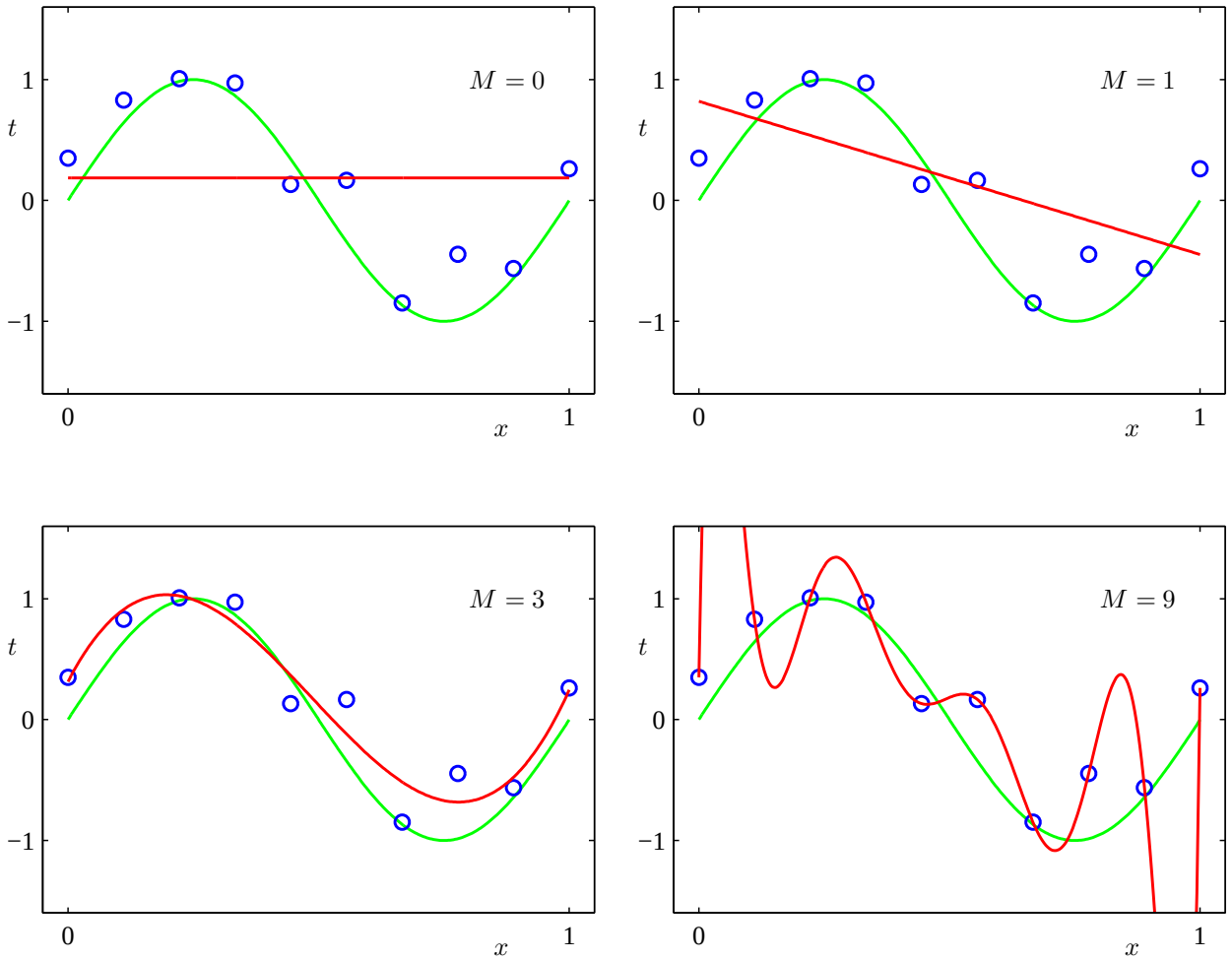
Pattern Recognition and Machine Learning (Fall 2022)

Outline I

Overfitting: k-NN



Overfitting: Regression



Training vs Testing

- Training data: $\mathcal{S} = \{(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(N)}, y^{(N)})\}$
- Test data: future observations that may be different from the training data
- Learning process:
  - learn prediction rule on training data
  - evaluate performance on test data
- Why separate training and testing?
  - training error is usually unrealistic low (overfitting)
  - error on test data (data not used to fit model) is more realistic

Training vs Testing (2)

- Goal: predict well on test data
- Two aspects:
  - model fit training data well
    - requires a more complex model
  - behavior of model on test data should match that on training data
    - requires a less complex (more stable) model
- Model complexity:
  - more complex model: smaller training error but larger difference between test and training error
  - less complex model: larger training error but smaller difference between test and training error

Regularization: A Technique Toward Overfitting

- Training error(f) = $\frac{1}{N}\sum_{i=1}^{N}(f(\mathbf{x}^{(i)}) - y^{(i)})^2$
- Test error(f) = $\mathbb{E}_{(\mathbf{x},y)}[f(\mathbf{x}) - y]^2$
- Model complexity: the difference between training and test error
  - increase model complexity decreases training error but increases difference between training and test
- Generalization: the ability to categorize correctly new samples that differ from those used for training.
- Regularization allows complex models to be trained on data sets of limited size without severe ovefitting, by limiting the effective model complexity.
- Benefits of regularization:
  - statistical: robust to large number of features
  - numerical: stabilize solution

Regularized Linear Regression

- Regularized error of least squares regression:

$$\frac{1}{2}\sum_{i=1}^{N}(\mathbf{w}^T\mathbf{x}^{(i)} - y^{(i)})^2 + \frac{\lambda}{2}\sum_{j=1}^{D}|w_j|^p$$

  - $\lambda \geq 0$ is the regularization parameter
  - $p = 0$: subset selection, non-convex, non-smooth
  - $p \in (0, 1)$: non-convex, smooth
  - $p \geq 1$: convex
  - $p = 1$: Lasso
  - $p = 2$: ridge repression
- Model complexity: small $\lambda \rightarrow$ large complexity

Ridge Regression

Regularization formulation:

$$\hat{\mathbf{w}}_{ridge} = \arg\min_{\mathbf{w}\in\mathbb{R}^D} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \sum_{j=1}^{D} |w_j|^2$$

.

- Implicit dimension reduction (the loadings can be recovered by regressing the principal component on D variable)
- More stable (smaller weight) than least squares
- It does not generally lead to sparse solution.
- Closely related to kernel method

Solution of Ridge Regression

- Denote $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}] \in \mathbb{R}^{D \times N}$,$\mathbf{y} = [y^{(1)}, \ldots, y^{(N)}] \in \mathbb{R}^{N}$.
- Solution of ridge regression:

$$\hat{\mathbf{w}}_{ridge} = (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I})^{-1}\mathbf{X}\mathbf{y}$$

.

- Compared to standard least square regression solution:

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

- Advantage: ridge regression allows $D > N$
  - stable: $\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I}$ is always invertible
  - implicit dimension reduction

Lasso

Regularization formulation:

$$\hat{\mathbf{w}}_{lasso} = \arg\min_{\mathbf{w}\in\mathbb{R}^D} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}^{(i)} - y^{(i)})^2 + \lambda \sum_{j=1}^{D} |w_j|$$

.

- Originally proposed in [Tib96], lasso for "least absolute shrinkage and selection operator".
- Convex optimization problem, but solution may not be unique.
- Global solution can be efficiently found (e.g., by Least Angle Regression (LARS) [EHJT04]).
- Solution is sparse, achieving feature selection.
- Solutionis not necessarily stable.

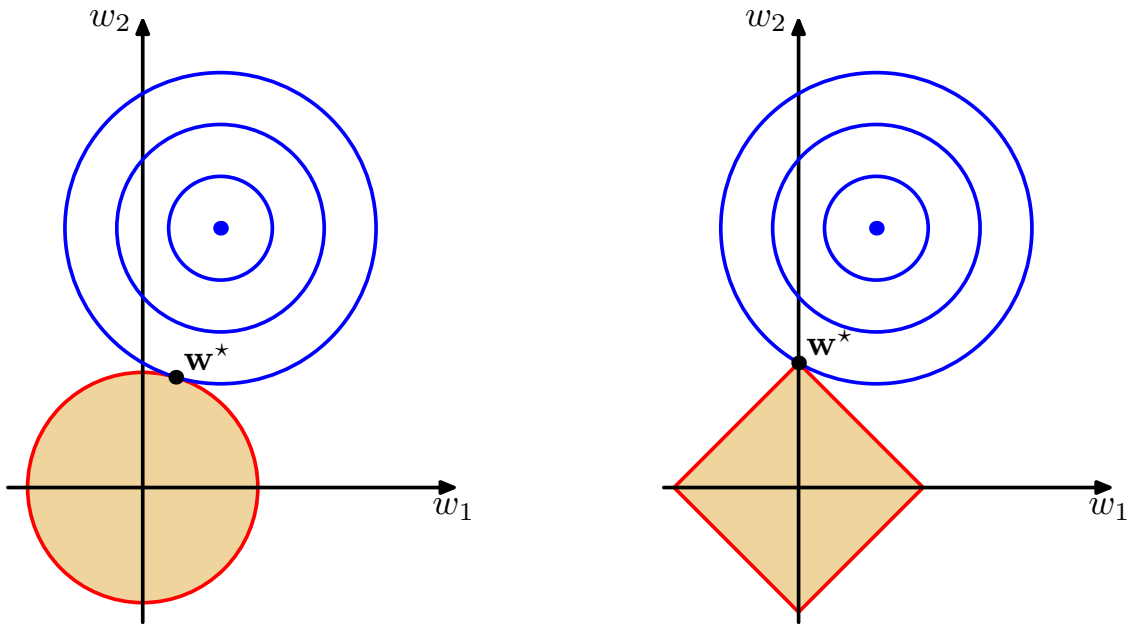
More on Lasso

- Limitation of lasso: the number of selected variables by the lasso is limited by the number of observations N.
  - E.g., for microarray data where there are thousands of genes but less than 100 samples
- Elastic net [ZH03]:

$$\hat{\mathbf{w}}_{en} = \arg\min_{\mathbf{w}\in\mathbb{R}^D} \sum_{i=1}^{N}(\mathbf{w}^T\mathbf{x}^{(i)} - y^{(i)}) + \lambda_2 \sum_{j=1}^{D} |w_j|^2 + \lambda_1 \sum_{j=1}^{D} |w_j|$$

  - Elastic net has grouping effect, i.e., it tends to select a group of highly correlated variables once one variable among them is selected.
  - Lasso tends to select only one out of the grouped variables and does not care which one is in the final model.

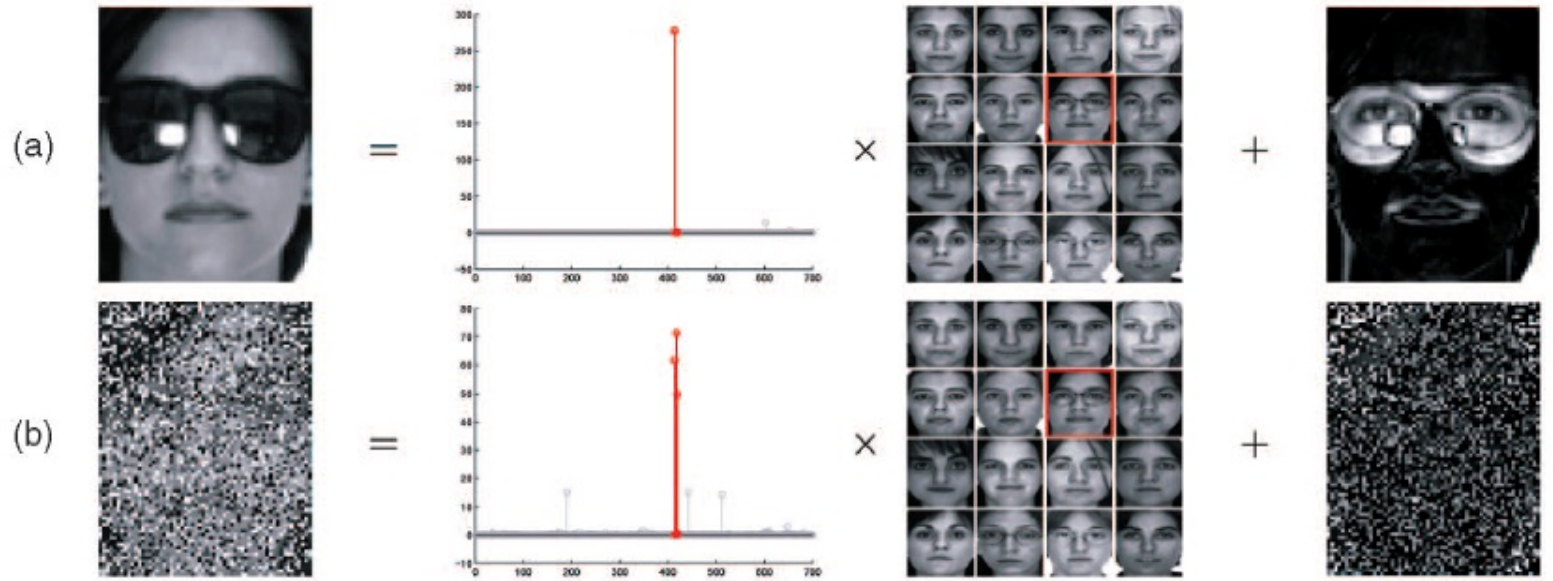
The effect of *L1* and *L2* Norms



- (left) Using the L_2 norm pulls directly towards the origin.
- (right) Using the L_1 norm pulls towards the coordinate axes, i.e., it tries to set some of the coordinates to 0.

Applications with Sparse Representation

- Robust face recognition



- Image super-resolution/inpainting/deblur
- Robust visual tracking
- More useful with lasso extensions:
  - tree-guided group lasso
  - graph-guided fused lasso
  - etc.

Generalization error

- Generalization error of a hypothesis: its expected error on examples not necessarily in the training set
- Informally,
  - Bias of a model is the expected generalization error even if we were to fit it to a very large training set.
  - Variance of a model is brought by the "spurious" samples in the training set.
  - "Simple" models with few parameters may have large bias (but small variance); "complex" models with many parameters may have large variance (but small bias).

Bias-variance Decomposition

- A frequentist viewpoint of the model complexity issue [GBD92][Fri97]
- Denote:
  - $f(\mathbf{x}; \mathcal{D})$ as the prediction function trained on data set $\mathcal{D}$
  - $h(\mathbf{x})$ as the optimal prediction
- Expected loss of least square regression:

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

  - $h(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}]$. Given unlimited supply of data, we could in principle find $h(\mathbf{x})$ to any degree of accuracy, which is the optimal choice of $f(\mathbf{x})$. For finite dataset, we do not know $h(\mathbf{x})$ exactly.
  - The second term, which is independent of $f(\mathbf{x})$, arises from the intrinsic noise on the data.

Bias-variance Decomposition (2)

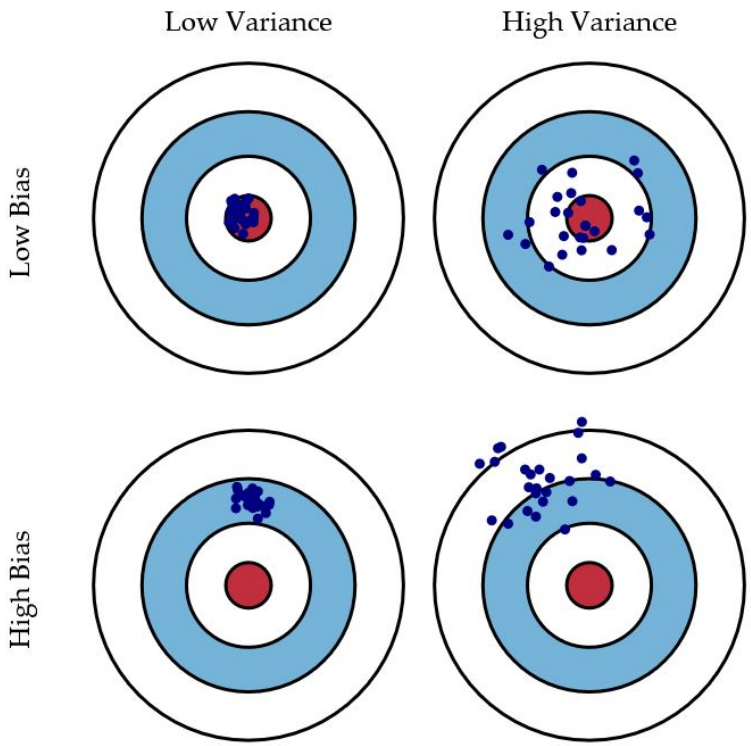
- For particular dataset $\mathcal{D}$:

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}}[(f(\mathbf{x};\mathcal{D}) - h(\mathbf{x}))^2] \\ = \; & (\mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})] - h(\mathbf{x}))^2 + \mathbb{E}_{\mathcal{D}}[(f(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})])^2] \\ = \; & (\mathit{bias})^2 + \mathit{variance} \end{aligned} \tag{1}$$

  - $\mathbb{E}_{\mathcal{D}}$ means take the average over the ensemble of data sets.
  - The squared bias represents the extent to which the average prediction over all data sets differs from the desired regression function.
  - The variance measures the extent to which the solutions for individual data sets vary around their average, the extent to which the function $f(\mathbf{x};\mathcal{D})$ is sensitive to the particular choice of data set.
- Expected loss = $(\text{bias})^2$ + variance + noise

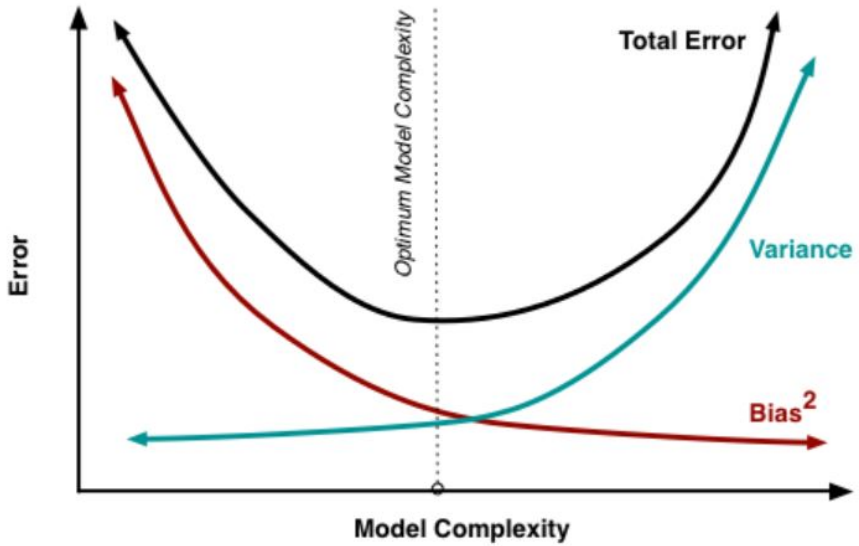
Intuitive Explanation

- Graphical illustration of bias and variance



Intuitive Explanation

- Bias and variance contributing to total error



An Example

- 100 independent data sets $\mathcal{D}^{(l)}, l = 1, \ldots, 100$, each containing $N = 25$ data points, generated from $\sin(2\pi x)$.
- Model:
  - Linear regression with 24 Gaussian basis functions

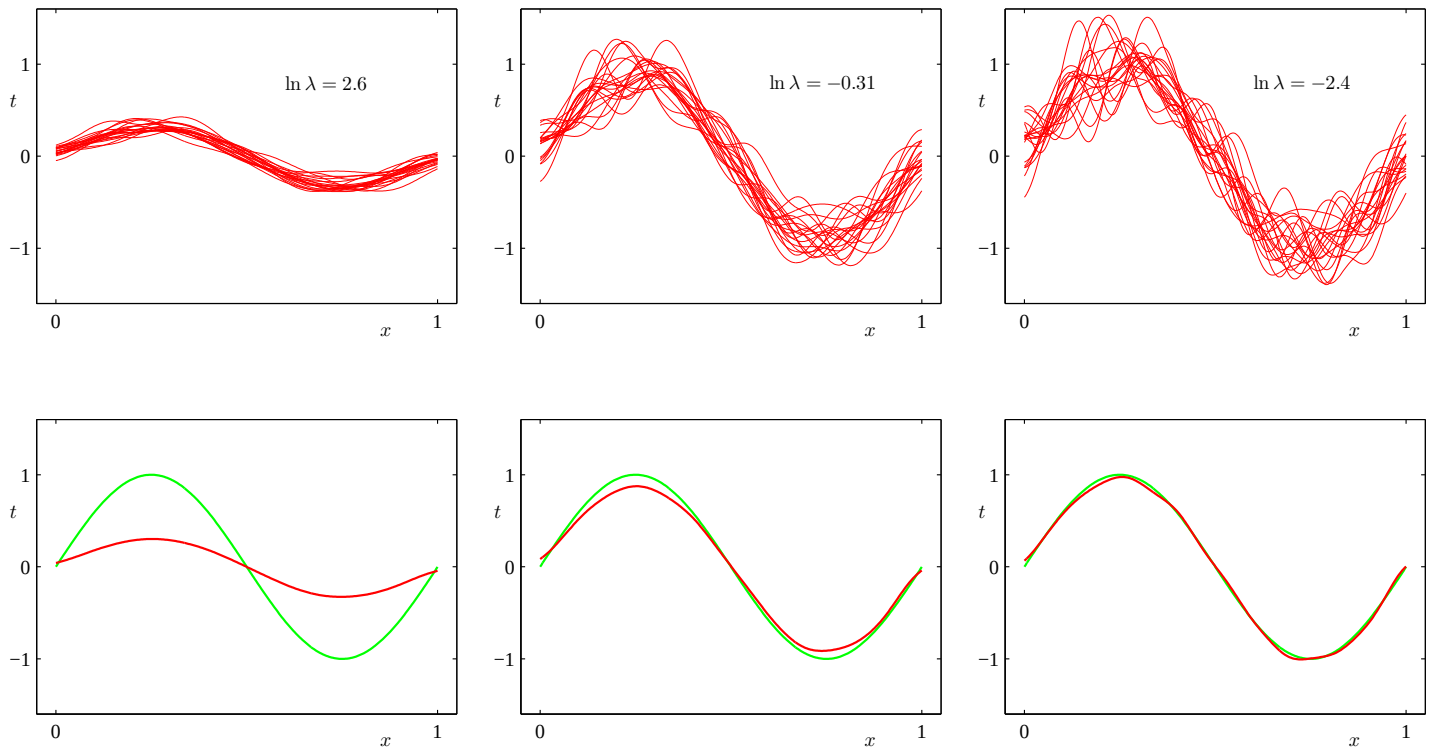
$$f(\mathbf{x}; \mathbf{w}) = w_0 + \sum_{j=1}^{24} w_j \phi_j(\mathbf{x}),$$

    where $\phi_j(\mathbf{x}) = \exp(-(\mathbf{x} - \boldsymbol{\mu}_j)^2/2s)$.
  - regularized least square error:

$$\sum_{i=1}^{N} (y^{(i)} - \mathbf{w}^T \Phi(\mathbf{x}^{(i)}))^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

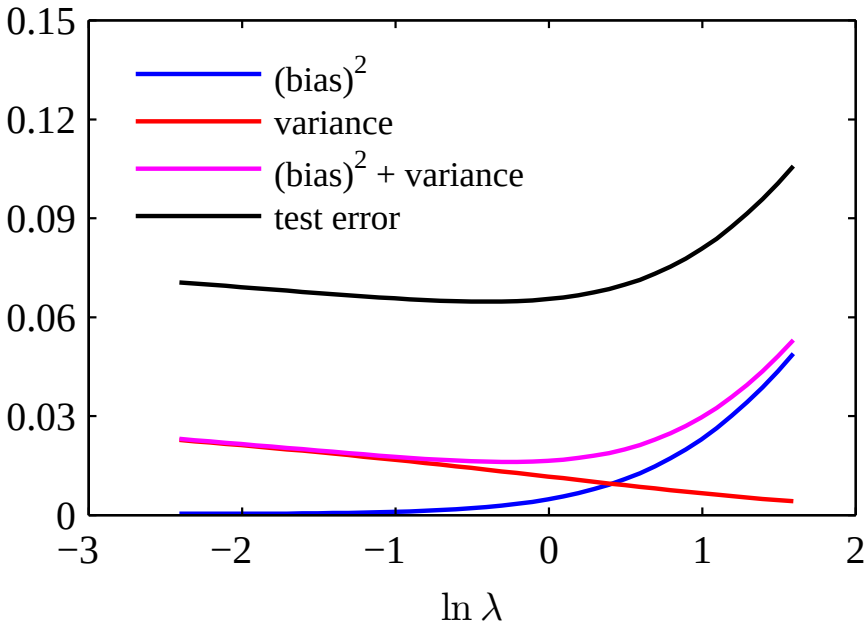
An Example

- Illustration of the dependence of bias and variance on model complexity, governed by regularization parameter λ



An Example

- Test set error for a test data set size of 1000 points



- Small λ allows the model to become finely tuned to the noise on each individual data set, leading to large variance.
- Large λ pulls the weight parameters towards zeros, leading to large bias.

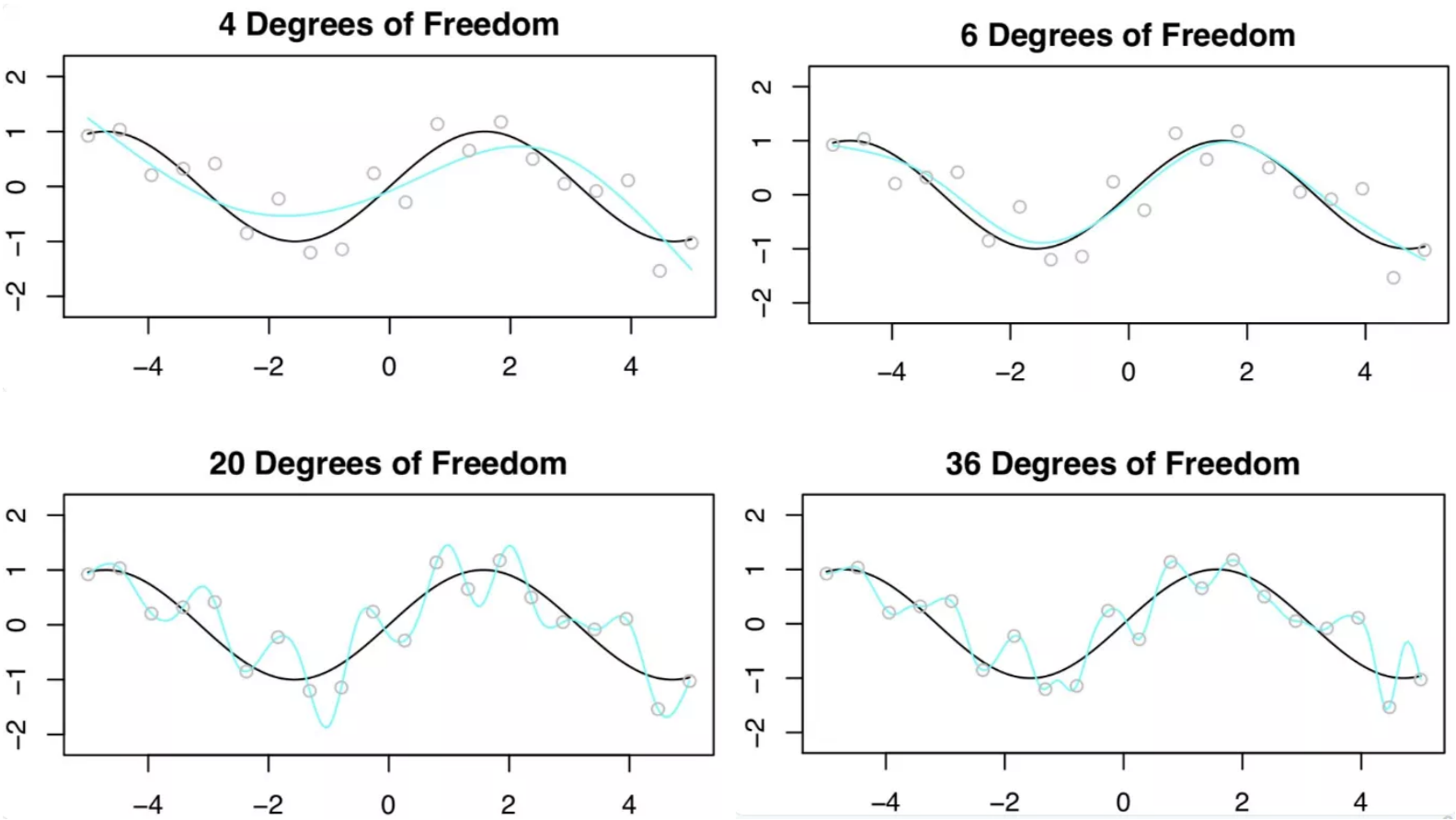
Bias-variance Tradeoff

- A good insight into model complexity issue:
  - Very flexible models having low bias and high variance
  - Relatively rigid models having high bias and low variance.
  - The model with the optimal predictive capability is the one that leads to the best balance between bias and variance.
- More discussions:
  - A bit of variance may be a good thing.
    - E.g., bias-corrected lease square estimator.
  - We sometimes would like to pay a little bias to save a lot of variance.
    - We can pay in model bias, or estimation bias.
    - Less rich model: model bias up, but less parameters to estimate (Feature selection)
    - Prefer well-behaved functions despite poorer fit: more estimation bias, lower variance (Regularization)

However ...

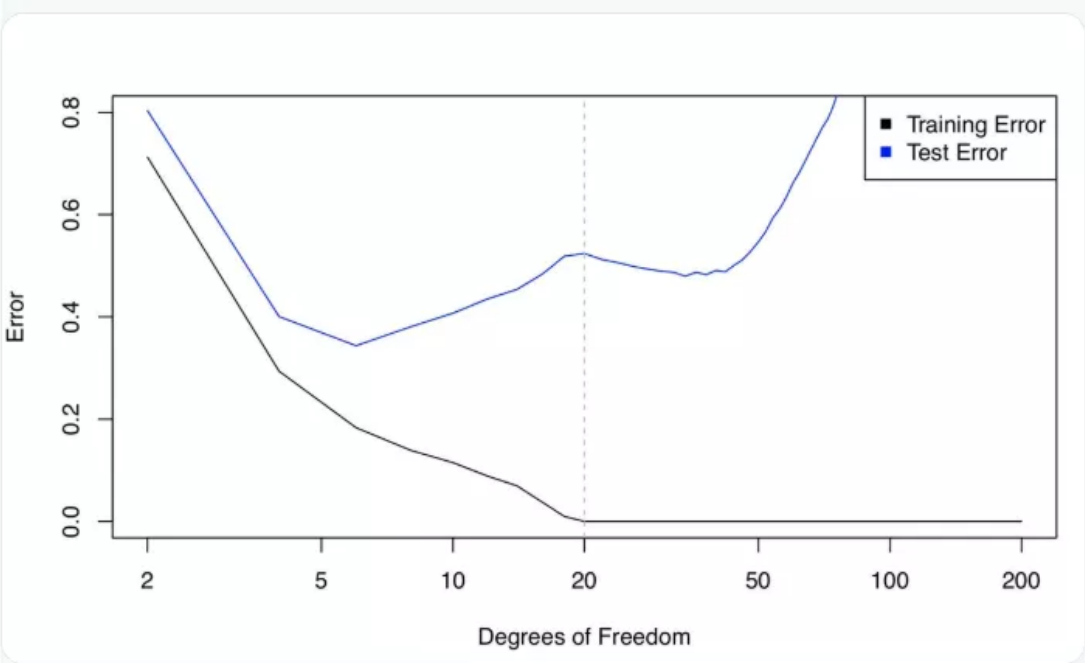
- Bias-variance decomposition has limited practical value
  - Bias and variance cannot be computed since it relies on knowing the true distribution of $\mathbf{x}$ and y(and hence $h(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}]$).
  - Bias-variance decomposition is based on averages with respect to ensembles of data sets, whereas in practice we have only the single observed data set.
- Bayesian model selection
  - Frequentist view: parameters are constant-valued but unknown
  - Bayesian view: parameters are random variables

Another Example



- Fitting with cubic spline, $N = 20$, degree of freedom = ♯ basis functions

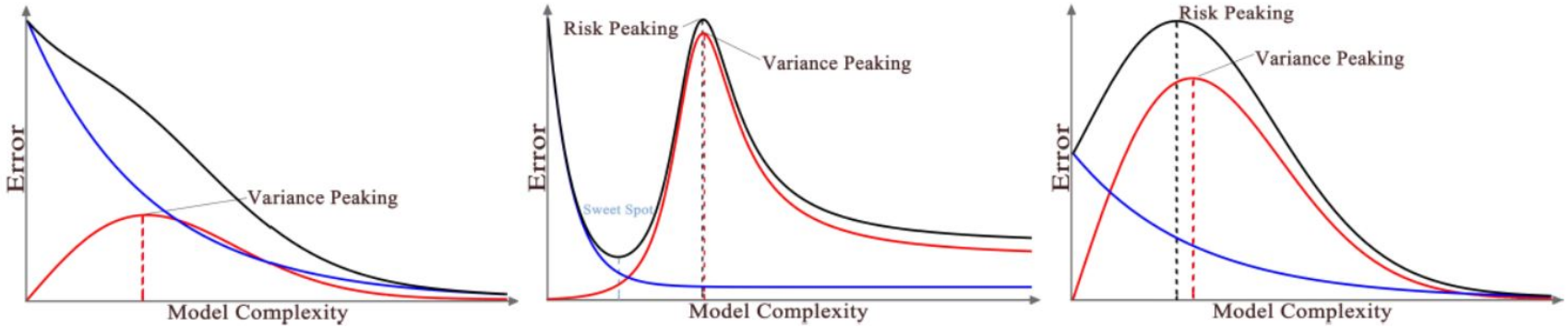
Double Descent Curve



- Why double descent curve should occur?
- Fitting with minimum norm

Generalization of Neural Networks

- Typical cases of expected risk curve (in black) in neural networks [ZY20].



  - squared bias, variance
  - The risk displays one of the three patterns: monotonically decreasing, double descent, and unimodal.
  - Label noise may increase the variance of the model and hence lead to double-descent risk.

A Few Questions

- How to explain bias variance tradeoff in statistical learning theory?
- How to relate training error to generalization error?
  - We care about generalization error,
  - but we fit models to the training set.
  - The goal is to estimate the difference of generalization error and training error.
- How good is the learning algorithm compared to the best possible prediction rule in a class?
- Are there conditions under which we can actually prove that the learning algorithms will work well?

Training and Generalization Error

- Training error (for binary classification):

$$\hat{\varepsilon}(f) = \frac{1}{N} \sum_{i=1}^{N} 1\{f(\mathbf{x}^{(i)}) \neq y^{(i)}\}$$

- Generalization error:

$$\varepsilon(f) = \mathbb{E}_{(\mathbf{x},y)}[1\{f(\mathbf{x}) \neq y\}]$$

  - Assumption: training and testing on the same distribution (one of PAC assumptions).
- Empirical risk minimization (ERM):
  - The most basic learning strategy
  - For $f_{\mathbf{w}}(\mathbf{x}) = 1\{\mathbf{w}^T\mathbf{x} \geq 0\}$, the ERM gives $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \hat{\varepsilon}(f_{\mathbf{w}})$.

ERM Over Hypothesis Class

- Hypothesis class $\mathcal{H}$ is the set of all classifiers considered by a learning algorithm.
  - For linear classification, $\mathcal{H} = \{f_{\mathbf{w}} : f_{\mathbf{w}} = 1\{\mathbf{w}^T\mathbf{x} \geq 0\}, \mathbf{w} \in \mathbb{R}^{D+1}\}$.
- Empirical risk minimization selects $\hat{f}$ to be the hypothesis with the smallest training error.

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \hat{\varepsilon}(f)$$

- How about the generalization error of $\hat{f}$?
  - $\hat{\varepsilon}(f)$ is a reliable estimate of $\varepsilon(f)$.
  - This implies an lower-bound on the generalization error of $\hat{f}$.

The Case of Finite H

- $\mathcal{H} = \{f_1, \ldots, f_k\}$
- For particular f_i, training error is close to generalization error with high probability, assuming N ($\sharp$ training examples) is large.

$$P(|\varepsilon(f_i) - \hat{\varepsilon}(f_i)| > \gamma) \leq 2e^{-2\gamma^2 N}$$

- For all $f \in \mathcal{H}$, we get uniform convergence result:

$$P(\forall f \in \mathcal{H}, |\varepsilon(f) - \hat{\varepsilon}(f)| \leq \gamma) \geq 1 - 2ke^{-2\gamma^2 N}$$

The Case of Finite H (2)

- Given $\gamma, \delta > 0$, if $N \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$, with probability at least $1 - \delta$, we have $|\varepsilon(f) - \hat{\varepsilon}(f)| \leq \gamma$ for all $f \in \mathcal{H}$.
  - setting $\delta = 2ke^{-2\gamma^2 N}$
  - definition of sample complexity bound
- Given N, δ, with probability at least $1 - \delta$, we have for all $f \in \mathcal{H}$,

$$|\varepsilon(f) - \hat{\varepsilon}(f)| \leq \sqrt{\frac{1}{2N} \log \frac{2k}{\delta}}.$$

The Case of Finite H (3)

- Define $f^* = \arg\min_{f \in \mathcal{H}} \varepsilon(f)$, the best possible hypothesis in $\mathcal{H}$
- Our learning algorithm picks $\hat{f} = \arg\min_{f \in \mathcal{H}} \hat{\varepsilon}(f)$.
- Comparing $\hat{f}$ with f^*:

$$\begin{aligned} \varepsilon(\hat{f}) &\leq \hat{\varepsilon}(\hat{f}) + \gamma \\ &\leq \hat{\varepsilon}(f^*) + \gamma \\ &\leq \varepsilon(f^*) + 2\gamma \end{aligned}$$

  - If uniform convergence holds, the generalization error of $\hat{f}$ is at most 2γ worse than the best possible hypothesis in $\mathcal{H}$.

Relation to Bias/Variance Tradeoff

Theorem

Given $|\mathcal{H}| = k, N$ and δ fixed, with probability at least $1 - \delta$, we have that

$$\varepsilon(\hat{f}) \leq (\min_{f \in \mathcal{H}} \varepsilon(f)) + 2\sqrt{\frac{1}{2N} \log \frac{2k}{\delta}}$$

- Suppose we switch from $\mathcal{H}$ to some larger hypothesis class $\mathcal{H}' \supseteq \mathcal{H}$, then
  - The first term decreases. The "bias" decreases.
  - The second term increases (with larger k). The "variance" increases.
- By holding γ and δ fixed and solving for N, we can get sample complexity bound.

VC Dimension

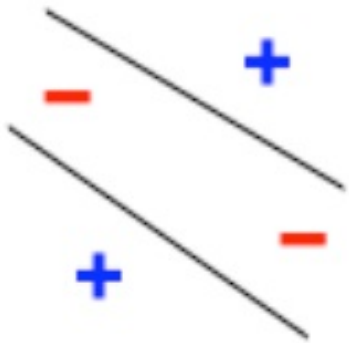
- Shattering: A function class $\mathcal{H}$ is said to shatter a set of data points $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})$ if $\mathcal{H}$ can realize any labeling on this data set, i.e., for every assignment of labels to those points $(y^{(1)}, \ldots, y^{(N)})$, there exists a function $f \in \mathcal{H}$ such that f makes no errors when evaluating that set of data points: $f(\mathbf{x}^{(i)}) = y^{(i)}$ for all i.

VC dimension

VC dimension $VC(\mathcal{H})$ is the size of the largest data set that can be shattered by $\mathcal{H}$.

VC Dimension - An Example

- Consider $\mathcal{H} = \{1(\mathbf{w}^T\mathbf{x} + b \geq 0)\}$ ($\mathbf{w} \in \mathbb{R}^2, b \in \mathbb{R}$), a set of binary linear classifiers in 2-D space.
- There exist 3 points $[0,0], [0,1], [1,0]$ that can be shattered by $\mathcal{H}$.
- Any four points cannot be shattered.



- So the VC dimension is 3.
- More general: D-dimensional linear classifier has VC dimension of $D+1$.

The Case of Infinite H

Theorem

Given $\mathcal{H}$, and let $D = VC(\mathcal{H})$. Then with probability at least $1 - \delta$, we have for all $f \in \mathcal{H}$,

$$|\varepsilon(f) - \hat{\varepsilon}(f)| \leq O(\sqrt{\frac{D}{N}\log\frac{N}{D} + \frac{1}{N}\log\frac{1}{\delta}}).$$

Thus, with probability at least $1 - \delta$, we also have

$$\varepsilon(\hat{f}) \leq \varepsilon(f^*) + O(\sqrt{\frac{D}{N}\log\frac{N}{D} + \frac{1}{N}\log\frac{1}{\delta}}).$$

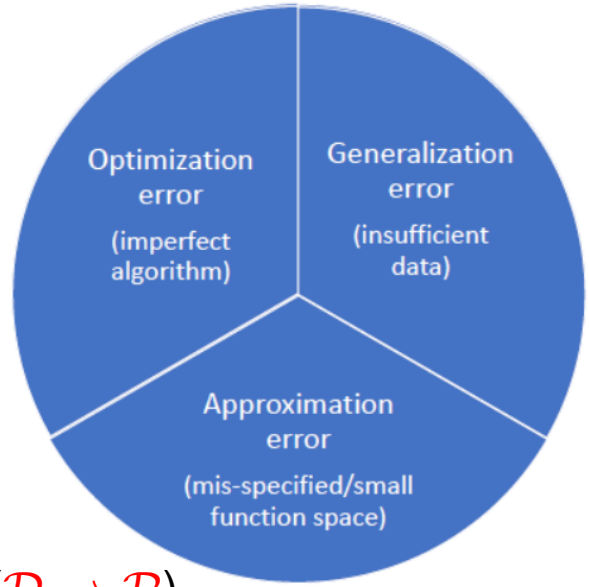
- If a hypothesis class has finite VC dimension, then uniform convergence occurs as N becomes large.

The Case of Infinite H (2)

- Corollary: For $|\varepsilon(f) - \hat{\varepsilon}(f)| \leq \gamma$ to hold for all $f \in \mathcal{H}$ (and hence $\varepsilon(\hat{f}) \leq \varepsilon(f^*) + 2\gamma$) with probability at least $1 - \delta$, it suffices that $N = O_{\gamma,\delta}(D)$.
  - The number of training examples needed to learn "well" using $\mathcal{H}$ is linear in the VC dimension of $\mathcal{H}$.
- For most hypothesis classes, the VC dimension is roughly linear in the number of parameters.
- Therefore, the number of training examples needed is roughly linear in the number of parameters of $\mathcal{H}$.

Statistical Learning Theory

- Reality: Find a function f from a hypothesis class $\mathcal{F}$ based on training data set $\mathcal{D}$
- Goal: f performs well on test data from a distribution $\mathcal{P}$



- Where is the gap:
  - generalization error ($\mathcal{D} \to \mathcal{P}$)
  - approximation error (Hypothesis space $\mathcal{F}$)
  - optimization error (How to find)

Decomposition of Learning Error

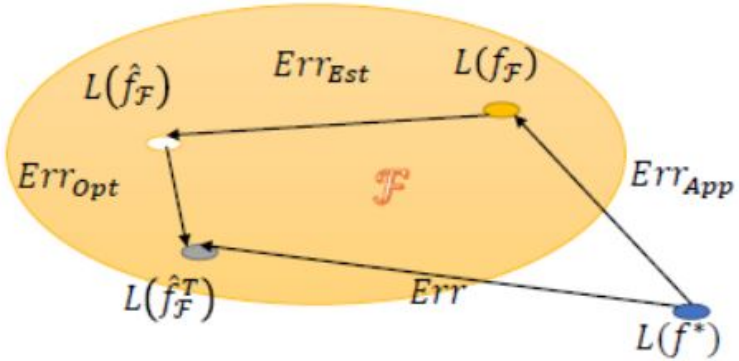
- Definition:
  - Training data: $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$
  - Model: $f \in \mathcal{F}$
  - Loss: $l(f(\mathbf{x}), y)$
  - Risk: $L(f) = \mathbb{E}[l(f(\mathbf{x}), y)]$
  - Empirical risk: $\hat{L}(f) = \frac{1}{N} \sum_{i=1}^{N} l(f(\mathbf{x}^{(i)}), y^{(i)})$
  - Bayes risk: $L^* = \min_f \mathbb{E}[l(f(\mathbf{x}), y)]$
  - Solutions:

$$
\begin{aligned}
f^* &= \arg\min_f \mathbb{E}[l(f(\mathbf{x}), y)] \quad \text{(Bayes prediction function)} \\
f_{\mathcal{F}} &= \arg\min_{f \in \mathcal{F}} \mathbb{E}[l(f(\mathbf{x}), y)] \\
\hat{f}_{\mathcal{F}} &= \arg\min_{f \in \mathcal{F}} \hat{L}(f) \quad \text{(ERM model)} \\
\hat{f}_{\mathcal{F}}^T & \quad \text{(model produced by the algorithm at } T\text{'s iteration)}
\end{aligned}
$$

Decomposition of Learning Error

Excess risk: Optimization Error Estimation Error Approximation Error

$$L(\hat{f}_{\mathcal{F}}^{T}) - L(f^{*}) = \left(L(\hat{f}_{\mathcal{F}}^{T}) - L(\hat{f}_{\mathcal{F}})\right) + \left(L(\hat{f}_{\mathcal{F}}) - L(f_{\mathcal{F}})\right) + \left(L(f_{\mathcal{F}}) - L(f^{*})\right)$$



Guarantees for Three Errors

- Optimization error
  - Convergence rate of optimization algorithm
  - $L(\hat{f}_{\mathcal{F}}^{T}) - L(\hat{f}_{\mathcal{F}}) \leq \epsilon(Alg, \mathcal{F}, N, T)$
- Estimation / generalization error:
  - Upper bound in terms of capacity
  - $L(\hat{f}_{\mathcal{F}}) - L(f_{\mathcal{F}}) \leq 2 \sup_{f \in \mathcal{F}} |\hat{L}(f) - L(f)| \leq \epsilon(Cap(\mathcal{F}), N)$
- Approximation error
  - Cannot be controllable in general
  - Universal approximation theorem of neural networks

Cross Validation

- Hold-out cross validation: split training set $\mathcal{D}$ into training set and validation set.
  - waste training data!
- k-fold cross validation
  - commonly used choice: $k = 10$
  - If data is really scarce, we may set $k = N$, which is leave-one-out cross validation.
  - bias-variance tradeoff
- Note that test set is never used in cross validation.

Feature Selection

- Given labeled data, we compute some scores that measures how informative each feature is about the class label.
- Ranking criteria:
  - mutual information
  - Bayes error
  - redundancy
- Feature select schemes:
  - filter: direct feature ranking
  - wrapper: determine the features based on performance under the learning algorithms to be used.
  - simultaneous learning and feature selection. E.g., lasso, Bayesian feature selection, etc.

Frequentist vs. Bayesian

- Frequentist: The parameters are constant-valued but unknown.
- Bayesian: The parameters are random variables whose values are unknown. We would specify prior distribution to express our "prior beliefs" about the parameters.
- In practice, Bayesian MAP estimate is less susceptible to overfitting than the ML estimate of the parameters, which also automatically determines model complexity using training data alone.
- An example: for text classification with $D \gg N$, Bayesian logistic regression turns out to be an effective algorithm.

Bayesian Model Selection

- Bayesian model selection is used when there exists prior knowledge about the appropriate class of approximating functions, represented as a prior distribution over models, $p(\text{model})$.
- Posterior distribution over models given the data:

$$p(\text{model}|\text{data}) = \frac{p(\text{data}|\text{model})p(\text{model})}{p(\text{data})}$$

- From the posterior distribution, we may:
  - choose the model with the highest posterior probability, or
  - choose multiple models with high posterior probabilities, or
  - use all models weighted by their posterior probabilities.
- Regularization may be seen as a Bayesian approach in which the prior favors simpler models.


B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani.
Least angle regression.
Annals of Statistics, 32(2):407–499, 2004.


J. H. Friedman.
On bias, variance, 0-1 loss, and the curse-of-dimensionality.
Data Mining and Knowledge Discovery, 1(1):55–77, 1997.


S. Geman, E. Bienenstock, and R. Doursat.
Neural networks and the bias/variance dilemma.
Neural Computation, 4(1):1–58, 1992.


R. Tibshirani.
Regression shrinkage and selection via the lasso.
Journal of the Royal Statistical Society, Series B, 58(1):267–288, 1996.


H. Zou and T. Hastie.
Regression shrinkage and selection via the elastic net, with applications to microarrays.
Technical report, Department of Statistics, Stanford University, 2003.

Chong You Jacob Steinhardt Yi Ma Zitong Yang, Yaodong Yu.
Rethinking bias-variance trade-off for generalization of neural networks.
In *Proceedings of International Conference on Machine Learning*, 2020.