CS:E4830 Kernel Methods in Machine Learning

Lecture 4 : Introductory Statistical Learning Theory

23rd March, 2022

Some Announcements

- Lecture slides of this (4th) lecture - uploaded to Mycourses
- Assignment 2 will be released by Friday (25th)
- There is no exercise session tomorrow

Generalization in Machine Learning

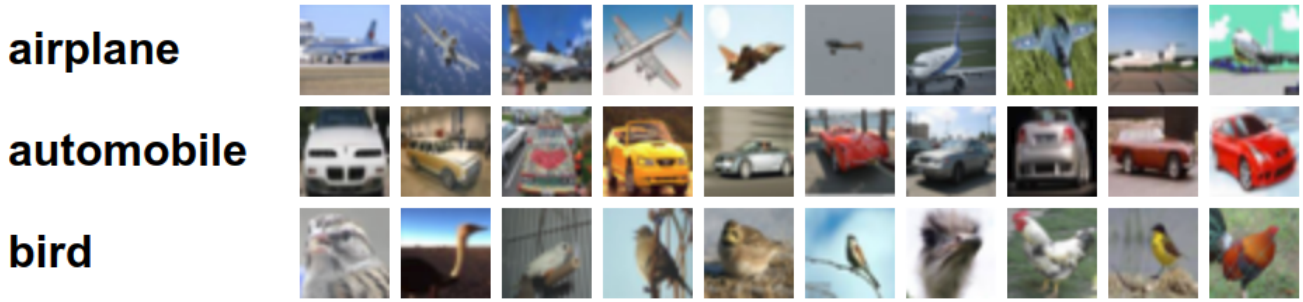

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

Generalization in Machine Learning

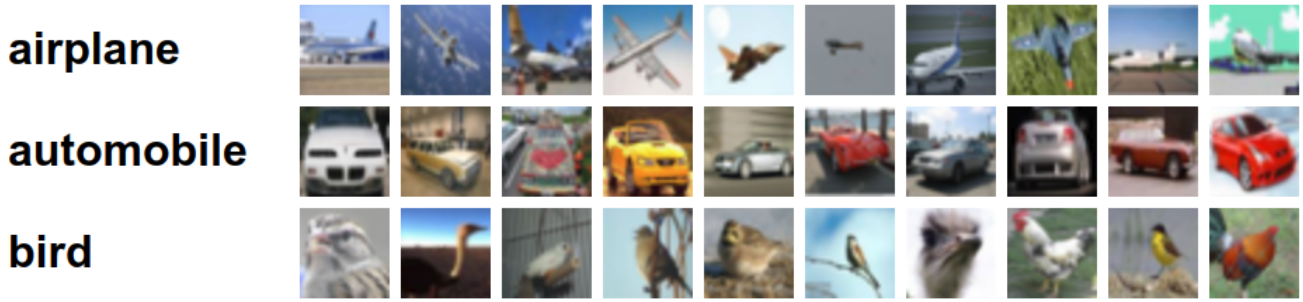

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

Consider the following scenario :

- Train a deep net on the above dataset, and test on the test set
  - What are the training error and test errors?

Generalization in Machine Learning

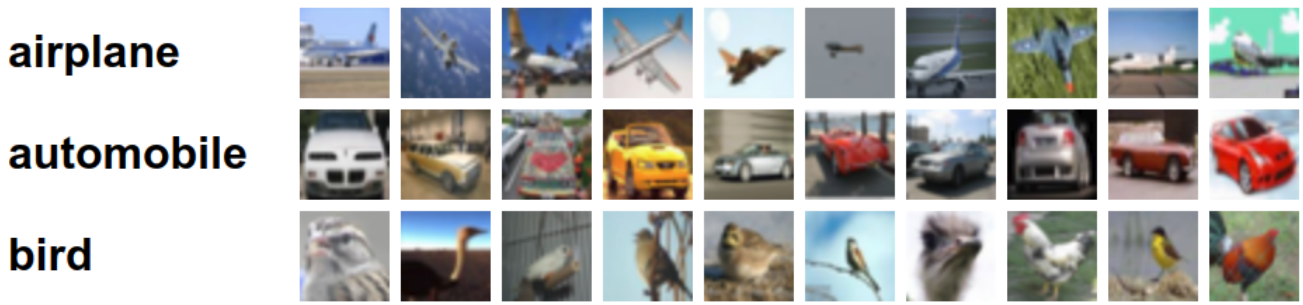

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

Consider the following scenario :

- Train a deep net on the above dataset, and test on the test set
  - What are the training error and test errors?
- Now, keep the training set images same but randomly shuffle their labels
  - Keep the test set same as the previous case
  - Train the deep net again on the training set with same images but this time with randomized labels, and test on the test set,
  - What are the training and test errors?

Generalization in Machine Learning

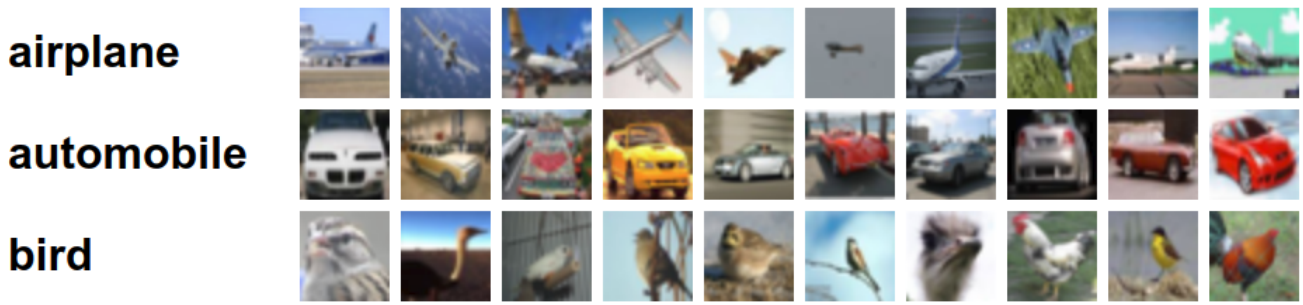

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

Consider the following scenario :

- Train a deep net on the above dataset, and test on the test set
  - What are the training error and test errors?
- Now, keep the training set images same but randomly shuffle their labels
  - Keep the test set same as the previous case
  - Train the deep net again on the training set with same images but this time with randomized labels, and test on the test set,
  - What are the training and test errors?
  - Does the training process take longer in this case ?

Understanding Deep Learning Requires Rethinking Generalization

- Understanding Deep Learning Requires Rethinking Generalization - Best paper award ICLR 2017

2.2 Implications

In light of our randomization experiments, we discuss how our findings pose a challenge for several traditional approaches for reasoning about generalization.

Rademacher complexity and VC-dimension. Rademacher complexity is commonly used and flexible complexity measure of a hypothesis class. The empirical Rademacher complexity of a hypothesis class $\mathcal{H}$ on a dataset $\{x_1, \ldots, x_n\}$ is defined as

$$\hat{\mathfrak{R}}_n(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right] \tag{1}$$

where $\sigma_1, \ldots, \sigma_n \in \{\pm 1\}$ are i.i.d. uniform random variables. This definition closely resembles our randomization test. Specifically, $\hat{\mathfrak{R}}_n(\mathcal{H})$ measures ability of $\mathcal{H}$ to fit random ± 1 binary label assignments. While we consider multiclass problems, it is straightforward to consider related binary classification problems for which the same experimental observations hold. Since our randomization tests suggest that many neural networks fit the training set with random labels perfectly, we expect that $\hat{\mathfrak{R}}_n(\mathcal{H}) \approx 1$ for the corresponding model class $\mathcal{H}$. This is, of course, a trivial upper bound on the Rademacher complexity that does not lead to useful generalization bounds in realistic settings. A similar reasoning applies to VC-dimension and its continuous analog fat-shattering dimension, unless we further restrict the network. While Bartlett (1998) proves a bound on the fat-shattering

Figure: Discussion in the paper

- Short youtube video `https://www.youtube.com/watch?v=fRndY_MoTlU`

Statistical Learning Theory - Goals

Goals of SLT

- Learnability - Which kinds of problems are learnable?
- Assumptions for learnability - What kinds of assumptions we need to make
- Algorithms - What are the performance guarantees of learning algorithms on unseen data

Basic setup of Statistical Learning Theory

Supervised binary classification

- Input $\mathcal{X}$, can be in various forms such as images, text documents and audio
- Output $\mathcal{Y} = \{-1, +1\}$ - binary classification for this lecture
- Other possible setups can be as follows :
  - One-hot encoded binary vector for multi-class classification - Cifar10
  - Multi-label classification - Wikipedia
- Joint probability distribution P over $\mathcal{X} \times \mathcal{Y}$
  - Training set $S = (x_i, y_i)_{i=1}^n$ consists of samples that are sampled independently and identically from this joint distribution P.
- The goal is to build a classifier f to predict the label $\hat{y}$ for a test instance x.

Assumptions of SLT

Assumptions

- Nothing is assumed a-priori on the nature of underlying data generating distribution P - (unlike in many cases where a distribution such as Gaussian is assumed)
- Labels can be noisy
  - Typically the labels annotations are provided by humans, who have different perspectives
- The distribution P over $\mathcal{X} \times \mathcal{Y}$ is fixed and does not change w.r.t. time
- Training points are sampled independently

The goal is not to estimate P, but predict the true label of test instances, and give guarantees on the test error of these predictors compared to the training error.

Important Terminology

- Loss of a classifier f on an input-output pair x, y. In this lecture, we will focus on 0-1 loss :

$$\ell(y, f(x)) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

Important Terminology

- Loss of a classifier f on an input-output pair x, y. In this lecture, we will focus on 0-1 loss :

$$\ell(y, f(x)) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

- Empirical error of classifier f is given by $R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$

Important Terminology

- Loss of a classifier f on an input-output pair x, y. In this lecture, we will focus on 0-1 loss :

$$\ell(y, f(x)) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

- Empirical error of classifier f is given by $R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$
- Expected loss of f

$$R(f) := \mathbb{E}_P(\ell(y, f(x)))$$

The above expectation is w.r.t the joint distribution P over $\mathcal{X} \times \mathcal{Y}$

Important Terminology

- Loss of a classifier f on an input-output pair x, y. In this lecture, we will focus on 0-1 loss :

$$\ell(y, f(x)) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

- Empirical error of classifier f is given by $R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$
- Expected loss of f

$$R(f) := \mathbb{E}_P(\ell(y, f(x)))$$

  The above expectation is w.r.t the joint distribution P over $\mathcal{X} \times \mathcal{Y}$
- Intuitively, $R_{emp}(f) \to R(f)$ as $n \to \infty$

Bayes Classifier

Bayes classifier f_{Bayes}, is defined to be the one which has the least classification error, i.e., $f_{Bayes} := \arg\min_f R(f) = \arg\min_f \mathbb{E}_P(\ell(y, f(x)))$

Bayes Classifier

Bayes classifier f_{Bayes}, is defined to be the one which has the least classification error, i.e., $f_{Bayes} := \arg\min_f R(f) = \arg\min_f \mathbb{E}_P(\ell(y, f(x)))$

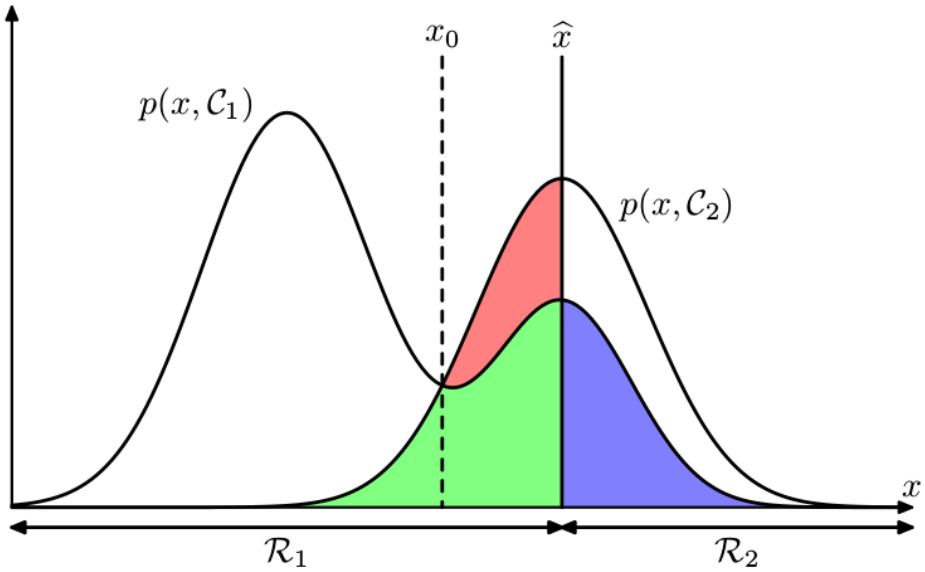

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

- Given a point on the x-axis, say $x = \hat{x}$, how do we compute $P(y = C_1|X = \hat{x})$ or $P(y = C_2|X = \hat{x})$?

Bayes Classifier

Bayes classifier f_{Bayes}, is defined to be the one which has the least classification error, i.e., $f_{Bayes} := \arg\min_f R(f) = \arg\min_f \mathbb{E}_P(\ell(y, f(x)))$

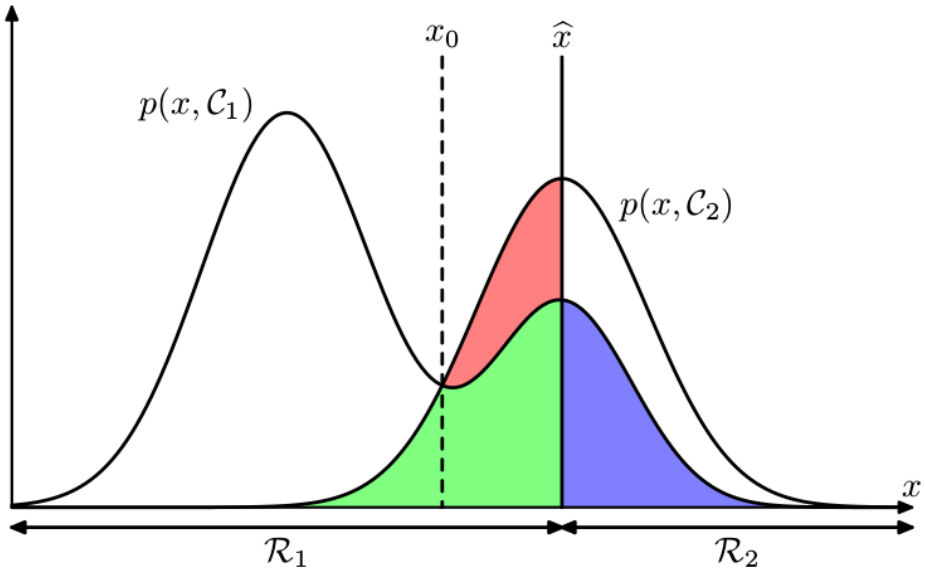

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

- Given a point on the x-axis, say $x = \hat{x}$, how do we compute $P(y = C_1|X = \hat{x})$ or $P(y = C_2|X = \hat{x})$?
- If the classifier is thresholded at $\hat{x}$, what kind of errors are signified by the red, green and blue regions?

Bayes Classifier

Bayes classifier f_{Bayes}, is defined to be the one which has the least classification error, i.e., $f_{Bayes} := \arg\min_f R(f) = \arg\min_f \mathbb{E}_P(\ell(y, f(x)))$

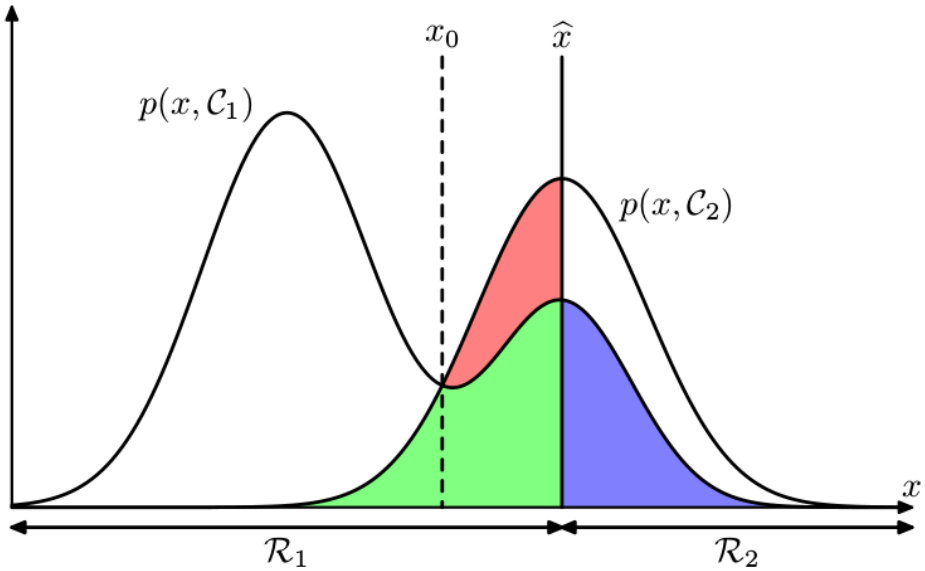

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

- Given a point on the x-axis, say $x = \hat{x}$, how do we compute $P(y = C_1|X = \hat{x})$ or $P(y = C_2|X = \hat{x})$?
- If the classifier is thresholded at $\hat{x}$, what kind of errors are signified by the red, green and blue regions?
- At what point in the graph $P(y = C_1|X = x) = 0.5$?

Notion of Generalization

It is desired that the error of our classifier is close to that of Bayes classifier, but this may not be possible due to the unknown underlying distribution. However, another desirable property for machine learning algorithms is that of **Generalization**

- Let f_n be a classifier obtained by some algorithm (such as deep net or SVM or Random forest) which is based on a finite training sample of size n.
- The classifier f_n generalizes well if the difference between empirical and expected error of f_n is low, i.e.,

$$|R(f_n) - R_{emp}(f_n)| \approx 0$$

- Note that having low generalization gap does not imply low expected or test error, it just means that **empirical error is a good indicator of expected error**

Overfitting

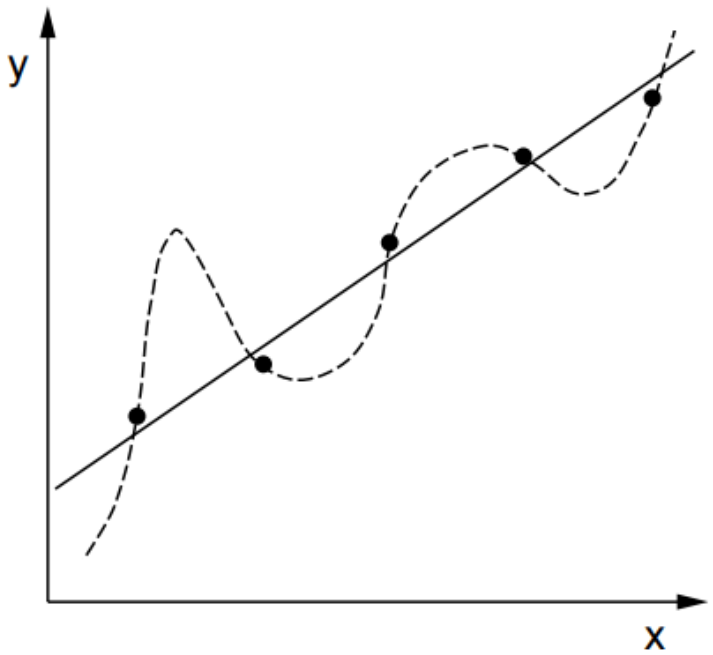

Figure: Overfitting example

Two of the many possible ways to fit the data (given by points in a regression setting)

- Complex model, a higher degree polynomial - no residual error
- Simpler linear model - has residual error

Components of classification error

Recall from the SLT framework, since we do not have access to the underlying data generating distribution. Therefore,

- We pick a function class $\mathcal{F}$ over which we find the best function that minimizes the error on training data.
- Based on your implementation, this function class can be :
  - Linear functions
  - Functions with bounded RKHS norms
  - Deep networks of certain depth

Components of classification error

Recall from the SLT framework, since we do not have access to the underlying data generating distribution. Therefore,

- We pick a function class $\mathcal{F}$ over which we find the best function that minimizes the error on training data.
- Based on your implementation, this function class can be :
  - Linear functions
  - Functions with bounded RKHS norms
  - Deep networks of certain depth
- Let's call the best function in the class $f_{\mathcal{F}}$, i.e., $f_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} R(f)$

Components of classification error

Recall from the SLT framework, since we do not have access to the underlying data generating distribution. Therefore,

- We pick a function class $\mathcal{F}$ over which we find the best function that minimizes the error on training data.
- Based on your implementation, this function class can be :
  - Linear functions
  - Functions with bounded RKHS norms
  - Deep networks of certain depth
- Let's call the best function in the class $f_{\mathcal{F}}$, i.e., $f_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} R(f)$
- Also, since we have finite training data, let the best function that we can find based on that data is f_n. Then,

$$R(f_n) - R(f_{Bayes}) = (R(f_n) - R(f_{\mathcal{F}})) + (R(f_{\mathcal{F}}) - R(f_{Bayes}))$$

- **Estimation error** (1st term) - $(R(f_n) - R(f_{\mathcal{F}}))$ - **finiteness of training data**
- **Approximation error** (2nd term) - $(R(f_{\mathcal{F}}) - R(f_{Bayes}))$ - **choice of function class**

Large vs Small Function class

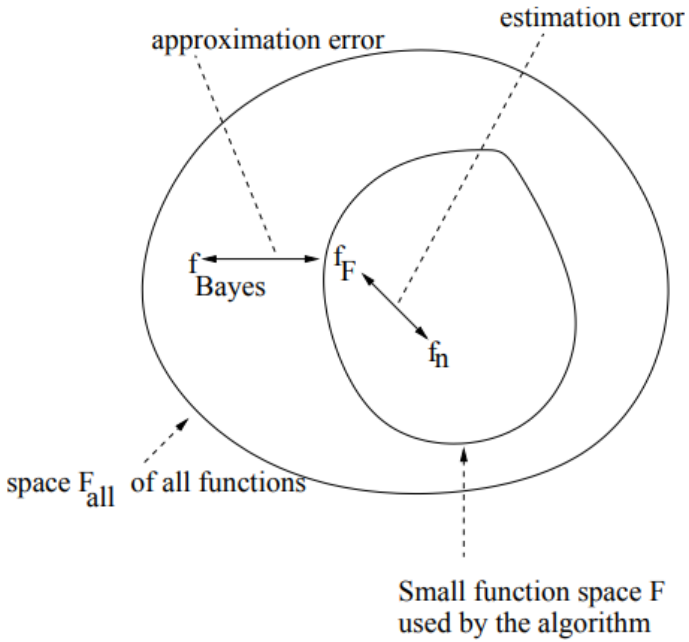

Figure: Pictorial depiction of the components of classification error

- The space F_{all} contains all possible functions that may be implmented using SVM, Deep nets, Random Forest and everything else
- **Estimation error** - $(R(f_n) - R(f_{\mathcal{F}}))$ - **finiteness of training data**
- **Approximation error** - $(R(f_{\mathcal{F}}) - R(f_{Bayes}))$ - **choice of function class**

Large vs Small Function class

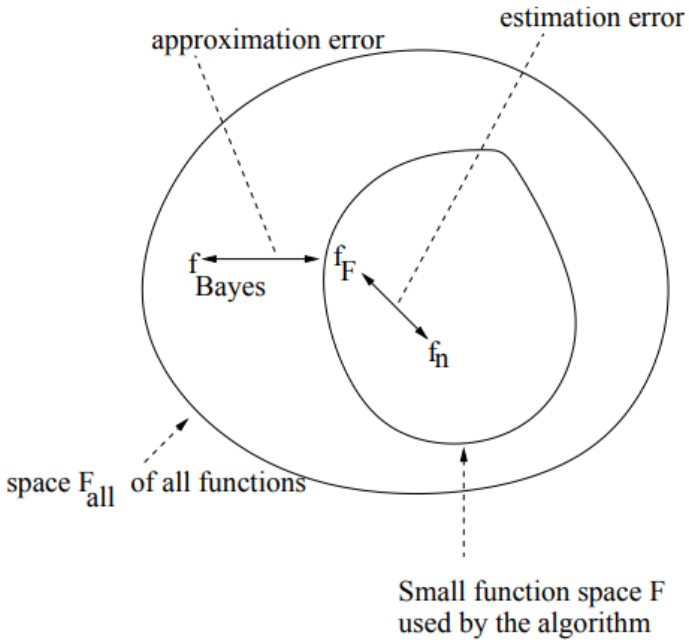

Figure: Pictorial depiction of the components of classification error

- The space F_{all} contains all possible functions that may be implmented using SVM, Deep nets, Random Forest and everything else
- **Estimation error** - $(R(f_n) - R(f_{\mathcal{F}}))$ - **finiteness of training data**
- **Approximation error** - $(R(f_{\mathcal{F}}) - R(f_{Bayes}))$ - **choice of function class**
- For example - If someone is claiming that using a deep net on a certain ML problem works better than a linear SVM, which of the two errors is actually going down?

Error variation with Function class capacity

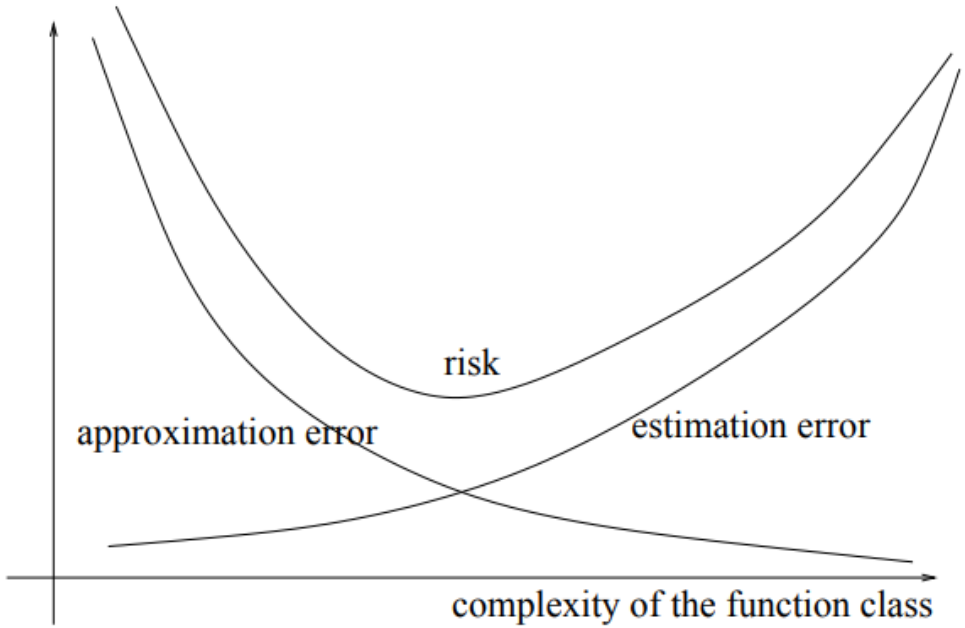

Figure: Variation of error components with the complexity of function class (tutorial by Von Luxburg and Schoelkopf)

- To the left with low complexity function class -
  - Linear classifier or kernel classifier with high bias or deep network with few layers
- To the right with high complexity function class -
  - kernel classifier with low bias or Deep neural network with many layers

Bias-Variance Tradeoff

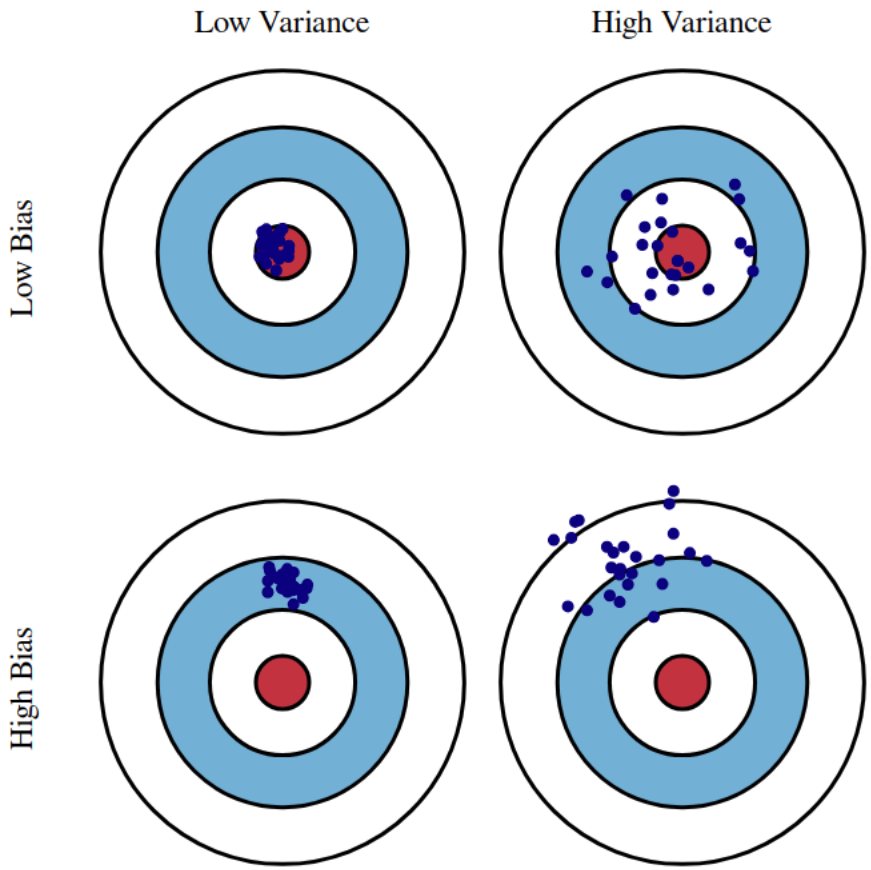

Figure: Pictorial depiction of the components of classification error

- **Approximation error** - $(R(f_{\mathcal{F}}) - R(f_{Bayes}))$ - corresponds to **Bias**
- **Estimation error** - $(R(f_n) - R(f_{\mathcal{F}}))$ - corresponds to **Variance**

Empirical Risk Minimization

In practice, learning algorithms (do not have access to the underlying data generating distribution P over $\mathcal{X} \times \mathcal{Y}$) are based on minimizing error on the training data. Formally, this is given as follows :

Principle of ERM

The idea behind the principle of Empirical Risk Minimization is to find a classifier in a pre-defined function class which minimizes the empirical risk. That is

$$f_n := \arg\min_{f \in \mathcal{F}} R_{emp}(f)$$

- We want to check if the classifier (function) f_n that we learn from **ERM is consistent or not** (defined on the next slide)?

Consistency of Learning Algorithm

Definition

Let $(x_i, y_i)_{i \in \mathbb{N}}$ be a sequence of training input-output pairs drawn according to some data distribution P. For each $n \in \mathbb{N}$, let f_n be the classifier that is learnt by some learning algorithm by seeing the first n training points, Then

- The learning algorithm (such as SVM and k-Nearest Neighbor) is called consistent w.r.t the function class $\mathcal{F}$ and the distribution P if the risk $R(f_n)$ converges in probability to the risk of the best possible classifier in $\mathcal{F}$

$$P(R(f_n) - R(f_{\mathcal{F}}) > \epsilon) \to 0 \text{ as } n \to \infty$$

- The motivation for the consistency of the principle of ERM comes from the law of large numbers.

Law of Large numbers

Let ξ_i be independent random variables drawn identically from a distribution P. Then the mean of the random variables converges to the mean of the distribution P when the sample size goes to infinity :

$$\frac{1}{n}\sum_{i=1}^{n}\xi_i \to \mathbb{E}(\xi) \text{ as } n \to \infty$$

Law of Large numbers

Let ξ_i be independent random variables drawn identically from a distribution P. Then the mean of the random variables converges to the mean of the distribution P when the sample size goes to infinity :

$$\frac{1}{n}\sum_{i=1}^{n}\xi_i \to \mathbb{E}(\xi) \text{ as } n \to \infty$$

- For ERM, let $\xi_i = \ell(f(x_i), y_i)$, then the law of large numbers gives the following :

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i)) \to E(\ell(y_i, f(x_i))) \text{ as } n \to \infty$$

- The above implies that the true risk (unknown due to the unknown probability distribution) can be approximated by the empirical risk (which can be computed from the training data)

Chernoff Bound

Non-asymptotic result

Chernoff Bound

Let ξ_i be independent random variables drawn identically from a distribution P. Then

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i - \mathbb{E}(\xi)\right| \geq \epsilon\right) \leq 2\exp(-2n\epsilon^2)$$

- The above inequality says that the probability that sample mean deviates from its expectation by ϵ goes down exponentially fast w.r.t sample size n

Chernoff Bound

Non-asymptotic result

Chernoff Bound

Let ξ_i be independent random variables drawn identically from a distribution P. Then

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i - \mathbb{E}(\xi)\right| \geq \epsilon\right) \leq 2\exp(-2n\epsilon^2)$$

- The above inequality says that the probability that sample mean deviates from its expectation by ϵ goes down exponentially fast w.r.t sample size n
- The same bound can be applied to empirical error and expected error of a classifier f. That is, for a **fixed function** f

$$P(|R_{emp}(f) - R(f)| \geq \epsilon) \leq 2\exp(-2n\epsilon^2)$$

- The above statement is a probabilistic argument, which means that it may not hold every time, and in fact, be violated in some cases (but with low probability)

Pictorial representation

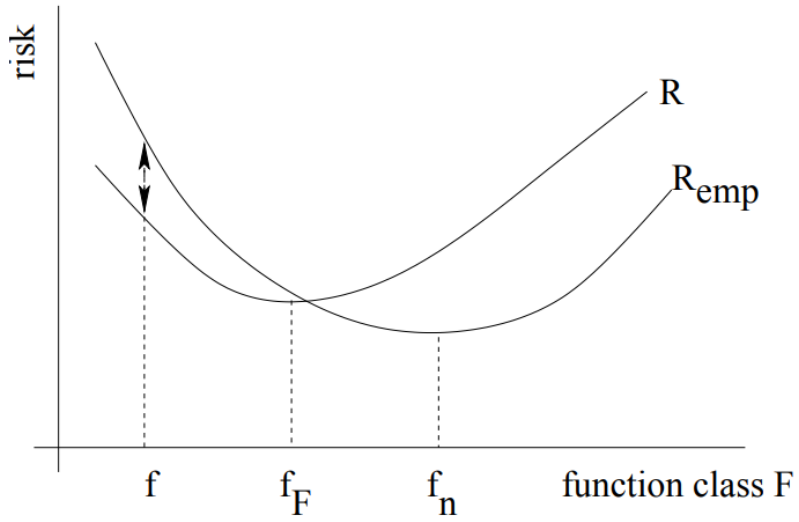

Figure: Depiction of training error and test error and various functions of interest

- The above picture shows variation of test error and training error (for a particular training set) as the function class capacity is increased

Pictorial representation

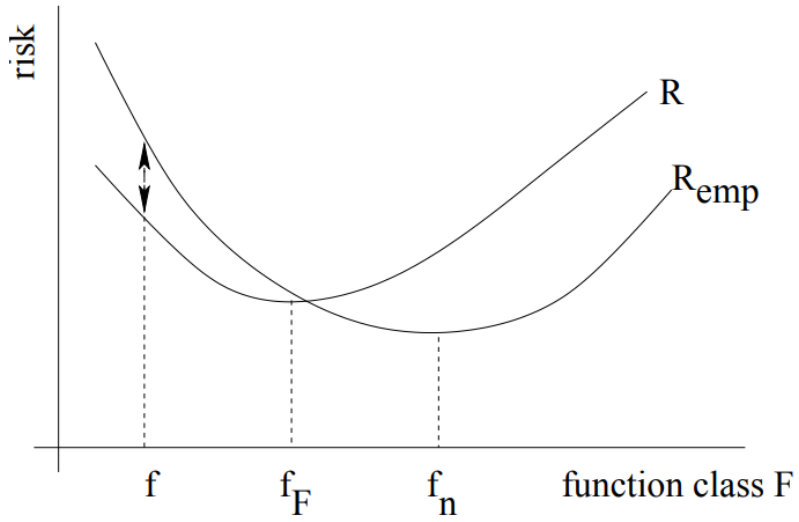

Figure: Depiction of training error and test error and various functions of interest

- The above picture shows variation of test error and training error (for a particular training set) as the function class capacity is increased
- As we make the training set larger, by Chernoff's bound, for **every fixed function**, $R_{emp}(f)$ converges to $R(f)$ (as shown by the double sided arrow),

Pictorial representation

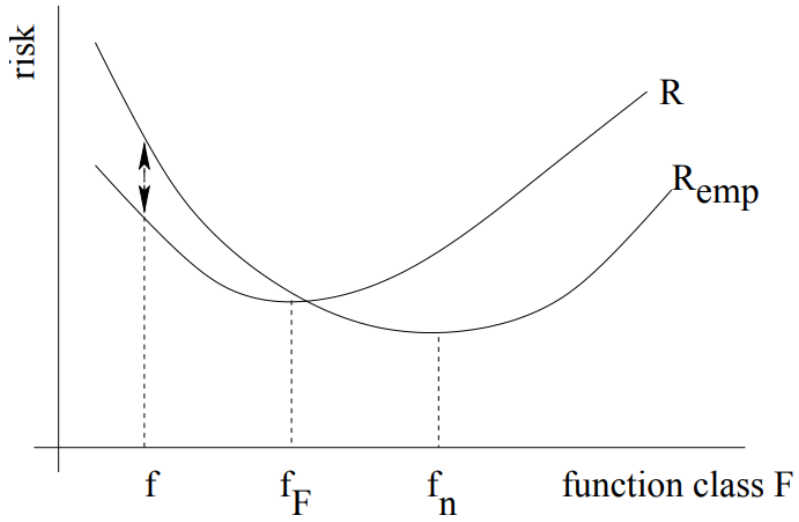

Figure: Depiction of training error and test error and various functions of interest

- The above picture shows variation of test error and training error (for a particular training set) as the function class capacity is increased
- As we make the training set larger, by Chernoff's bound, for **every fixed function**, $R_{emp}(f)$ converges to $R(f)$ (as shown by the double sided arrow),
- However, the above bound holds for a fixed function, which is not the case for ERM, which returns a different function **depending on training data**

Pictorial representation

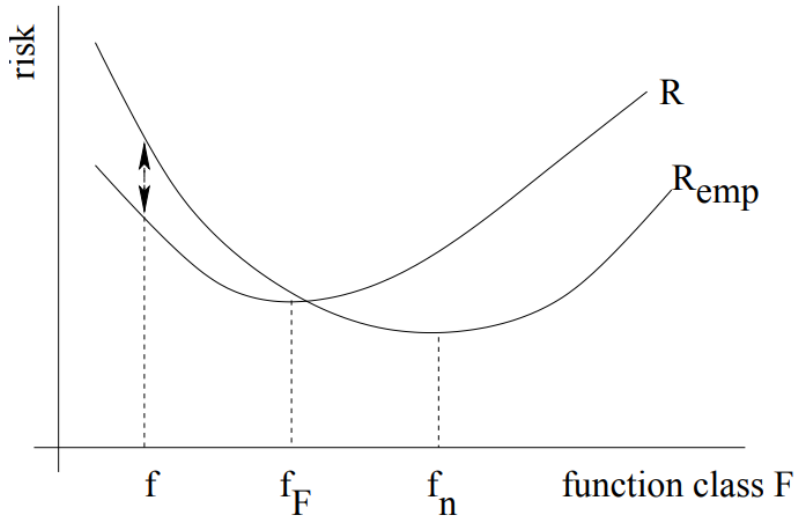

Figure: Depiction of training error and test error and various functions of interest

- The above picture shows variation of test error and training error (for a particular training set) as the function class capacity is increased
- As we make the training set larger, by Chernoff's bound, for **every fixed function**, $R_{emp}(f)$ converges to $R(f)$ (as shown by the double sided arrow),
- However, the above bound holds for a fixed function, which is not the case for ERM, which returns a different function **depending on training data**
- Therefore, it is **not guaranteed** that $R(f_n)$ converges to $R(f_F)$

Uniform Convergence - I

- **Uniform Convergence** is a condition over a function class which ensures consistency of ERM, and is given by $|R_{emp}(f) - R(f)| < \epsilon, \forall f \in \mathcal{F}$ for some finite sample size n
- Alternatively, the condition of Uniform Convergence can be stated $\sup_{f \in \mathcal{F}} |R_{emp}(f) - R(f)| < \epsilon$
- That is, for all functions $f \in \mathcal{F}$, the difference $|R_{emp}(f) - R(f)|$ becomes small **simultaneously**

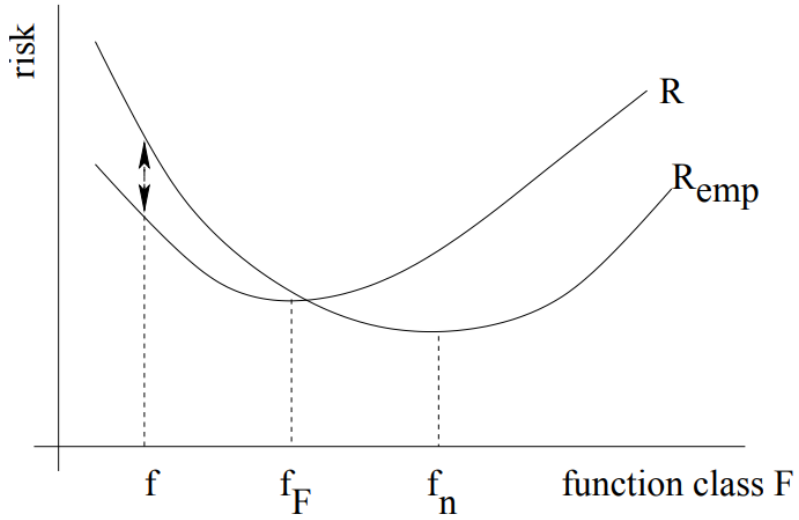

Figure: Under Uniform Convergence, the difference between the two curves becomes arbitrarily small for some large but finite sample size n

Uniform Convergence - II

We will show that if Uniform Convergence holds for a function class $\mathcal{F}$, then the Empirical Risk Minimizer is guaranteed to be consistent, i.e., $R(f_n) \to R(f_{\mathcal{F}})$ as $n \to \infty$. (Proof on the next slide)
First, couple of inequalities for the proof :

- The following holds (by definition of supremum/maximum), for any function $f \in \mathcal{F}$

$$|R(f) - R_{emp}(f)| \leq \sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)|$$

- Therefore, it also holds for the Empirical Risk Minimizer f_n which is chosen based on finite number (n) of samples

$$P(|R(f_n) - R_{emp}(f_n)| \geq \epsilon) \leq P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon)$$

Uniform Convergence - III

Sufficiency of Uniform Convergence for consistency of ERM

$$|R(f_n) - R(f_{\mathcal{F}})|$$
$$= R(f_n) - R(f_{\mathcal{F}}) \text{ (Why?)}$$

Uniform Convergence - III

Sufficiency of Uniform Convergence for consistency of ERM

$$
\begin{aligned}
&|R(f_n) - R(f_{\mathcal{F}})| \\
&= R(f_n) - R(f_{\mathcal{F}}) \text{ (Why?)} \\
&= R(f_n) - R_{emp}(f_n) + R_{emp}(f_n) - R_{emp}(f_{\mathcal{F}}) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}})
\end{aligned}
$$

Uniform Convergence - III

Sufficiency of Uniform Convergence for consistency of ERM

$$
\begin{aligned}
&|R(f_n) - R(f_{\mathcal{F}})| \\
&= R(f_n) - R(f_{\mathcal{F}}) \text{ (Why?)} \\
&= R(f_n) - R_{emp}(f_n) + R_{emp}(f_n) - R_{emp}(f_{\mathcal{F}}) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}}) \\
&\leq R(f_n) - R_{emp}(f_n) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}}) \text{ (Why?)}
\end{aligned}
$$

Uniform Convergence - III

Sufficiency of Uniform Convergence for consistency of ERM

$$
\begin{aligned}
&|R(f_n) - R(f_{\mathcal{F}})| \\
&= R(f_n) - R(f_{\mathcal{F}}) \text{ (Why?)} \\
&= R(f_n) - R_{emp}(f_n) + R_{emp}(f_n) - R_{emp}(f_{\mathcal{F}}) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}}) \\
&\leq R(f_n) - R_{emp}(f_n) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}}) \text{ (Why?)} \\
&\leq 2 \sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)|
\end{aligned}
$$

Uniform Convergence - III

Sufficiency of Uniform Convergence for consistency of ERM

$$
\begin{aligned}
&|R(f_n) - R(f_{\mathcal{F}})| \\
&= R(f_n) - R(f_{\mathcal{F}}) \text{ (Why?)} \\
&= R(f_n) - R_{emp}(f_n) + R_{emp}(f_n) - R_{emp}(f_{\mathcal{F}}) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}}) \\
&\leq R(f_n) - R_{emp}(f_n) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}}) \text{ (Why?)} \\
&\leq 2 \sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)|
\end{aligned}
$$

Therefore, $P(|R(f_n) - R(f_{\mathcal{F}})| \geq \epsilon) \leq P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon/2)$

Uniform Convergence - III

Sufficiency of Uniform Convergence for consistency of ERM

$$
\begin{aligned}
&|R(f_n) - R(f_{\mathcal{F}})| \\
&= R(f_n) - R(f_{\mathcal{F}}) \text{ (Why?)} \\
&= R(f_n) - R_{emp}(f_n) + R_{emp}(f_n) - R_{emp}(f_{\mathcal{F}}) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}}) \\
&\leq R(f_n) - R_{emp}(f_n) + R_{emp}(f_{\mathcal{F}}) - R(f_{\mathcal{F}}) \text{ (Why?)} \\
&\leq 2 \sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)|
\end{aligned}
$$

Therefore, $P(|R(f_n) - R(f_{\mathcal{F}})| \geq \epsilon) \leq P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon/2)$

- Since uniform law of large numbers holds for a function class $\mathcal{F}$ by uniform convergence, the RHS tends to 0
- Since the LHS is upper bounded by RHS, this implies the consistency of ERM over the function class.

Necessary & Sufficient Condition for consistency of ERM

- The previous proof shows that uniform convergence is a sufficient condition for the consistency of ERM

Necessary & Sufficient Condition for consistency of ERM

- The previous proof shows that uniform convergence is a sufficient condition for the consistency of ERM
- Is it also necessary?

Necessary & Sufficient Condition for consistency of ERM

- The previous proof shows that uniform convergence is a sufficient condition for the consistency of ERM
- Is it also necessary?

Theorem by Vapnik and Chervonenkis

Uniform convergence, i.e.,

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| > \epsilon) \to 0 \text{ as } n \to \infty$$

$\forall \epsilon > 0$ is a necessary and sufficient condition for consistency of ERM with respect to the function class $\mathcal{F}$.

NASC for consistency of ERM - II

Theorem by Vapnik and Chervonenkis

Uniform convergence, i.e.,

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| > \epsilon) \to 0 \text{ as } n \to \infty$$

$\forall \epsilon > 0$ is a necessary and sufficient condition for consistency of ERM with respect to the function class $\mathcal{F}$.

- Learning with all possible functions
  - Larger the function class $\mathcal{F}$, so is $|R(f) - R_{emp}(f)|$, and hence difficult to achieve consistency
- Learning with a restricted function class
  - On the contrary, small $\mathcal{F}$ means easier to learn consistent classifiers
- However, unfortunately, it is not easy to find out if the *uniform convergence holds for a function class or not*

Capacity of Function Class

The main quantity of interest from the previous theorem is the following :

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| > \epsilon)$$

- Can we study the above quantity in the non-asymptotic regime, i.e. when the sample size n is finite
  - Practically, this also matters more since we normally have finite data size

Capacity of Function Class

The main quantity of interest from the previous theorem is the following :

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| > \epsilon)$$

- Can we study the above quantity in the non-asymptotic regime, i.e. when the sample size n is finite
  - Practically, this also matters more since we normally have finite data size
- In bounding the quantity $P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon)$, there are two challenges :
  - Infinitely many functions, due to **continuous nature of the function class**
  - The expected risk $R(f)$, which depends on the underlying probability distribution, and **cannot be computed from training data**

Capacity of Function Class

The main quantity of interest from the previous theorem is the following :

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| > \epsilon)$$

- Can we study the above quantity in the non-asymptotic regime, i.e. when the sample size n is finite
  - Practically, this also matters more since we normally have finite data size
- In bounding the quantity $P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon)$, there are two challenges :
  - Infinitely many functions, due to **continuous nature of the function class**
  - The expected risk $R(f)$, which depends on the underlying probability distribution, and **cannot be computed from training data**
- To get a handle on this, we need the following three concepts :
  - Union bound
  - Symmetrization
  - Shattering

Union Bound

For the sake of simplicity to start with

Suppose there are m functions in the function class $\mathcal{F}$

$$
\begin{aligned}
& P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon) \\
& = P((|R(f_1) - R_{emp}(f_1)| \geq \epsilon) \text{ or } \ldots \text{ or } (|R(f_m) - R_{emp}(f_m)| \geq \epsilon)) \\
& \leq \sum_{i=1}^{m} P(|R(f_i) - R_{emp}(f_i)| \geq \epsilon) \text{ (result on the probability of union of events)} \\
& \leq 2m \exp(-2n\epsilon^2) \text{ (application of Chernoff bound)}
\end{aligned}
$$

- Thus, if a function class $\mathcal{F}$ has finite number of functions, then we can bound the probability that for some $f \in \mathcal{F}$, the difference between empirical error and expected error is greater than ϵ.

In practice, useful Function classes have infinitely many functions

- Now, how do we bound the original quantity $P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon)$ when there are infinitely many functions
- We have infinitely many functions when learning with SVMs or Deep networks :
  - Each of the many possible orientations of an SVM hyper-plane
  - Each of the many possible settings of weights of hidden units in a deep net
- The above challenge of infinite number of functions in bounding $P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon)$ is handled using two concepts
  - Symmetrization
  - Shattering

Symmetrization

Symmetrization Lemma

For $n\epsilon^2 \geq 2$

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| > \epsilon) \leq 2P(\sup_{f \in \mathcal{F}} |R_{emp}(f) - R'_{emp}(f)| > \epsilon/2)$$

where $R_{emp}(f) - R'_{emp}(f)$ refers to the difference between the empirical errors of two samples of size n, where first sample is the training sample, and the second one is called *ghost sample!* (since it does not exist in practice)

- The above lemma bounds $P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon)$ with **something that completely depends on training data** (we don't need to compute the expected error $R(f)$)
- **In simple words**, symmetrization lemma says - if the difference between empirical error of two independent samples is small (RHS of the above inequality), then the difference between empirical (training) error and expected error is also small (LHS of the above inequality).

Shattering Coefficient - I

Now we only need to focus on the following quantity :

$$2P(\sup_{f \in \mathcal{F}} |R_{emp}(f) - R'_{emp}(f)| > \epsilon/2)$$

Shattering Coefficient - I

Now we only need to focus on the following quantity :

$$2P(\sup_{f \in \mathcal{F}} |R_{emp}(f) - R'_{emp}(f)| > \epsilon/2)$$

- An interesting thing to note is that even though the number of functions $f \in \mathcal{F}$ are infinite, for a training set of size n, since each instance can have label from $\{+1, -1\}$, there are atmost 2^n different possible labelings

Shattering Coefficient - I

Now we only need to focus on the following quantity :

$$2P(\sup_{f \in \mathcal{F}} |R_{emp}(f) - R'_{emp}(f)| > \epsilon/2)$$

- An interesting thing to note is that even though the number of functions $f \in \mathcal{F}$ are infinite, for a training set of size n, since each instance can have label from $\{+1, -1\}$, there are atmost 2^n different possible labelings

Shattering Coefficient - II

For the training data, $Z_n := \{(x_i, y_i)\}_{i=1}^n$, denote by $|\mathcal{F}_{Z_n}|$, the number of functions in $\mathcal{F}$ that can be distinguished from their values on $\{x_1, \ldots x_n\}$.

Shattering Coefficient of $\mathcal{F}$ for sample size n

Shattering co-efficient of a function class counts the maximum number of possible labelings it can exhibit on **some sample** of size n. It is defined as follows :

$$\mathcal{N}(\mathcal{F}, n) := \max\{|\mathcal{F}_{Z_n}| \text{ such that } x_1, \ldots, x_n \in \mathcal{X}\}$$

- The maximum number of ways in which some sample of size n can be classified into two different classes.

Shattering - 1-D Pictorial representation

When $\mathcal{N}(\mathcal{F}, n) = 2^n$, then the function class $\mathcal{F}$ is said to **shatter** n points, that **there exists** a sample of n points, on which it can achieve all possible labelings. If the function class $\mathcal{F}$ consists of linear classifiers only, then :

- How many can be shattered by linear classifiers in 1-dimension?

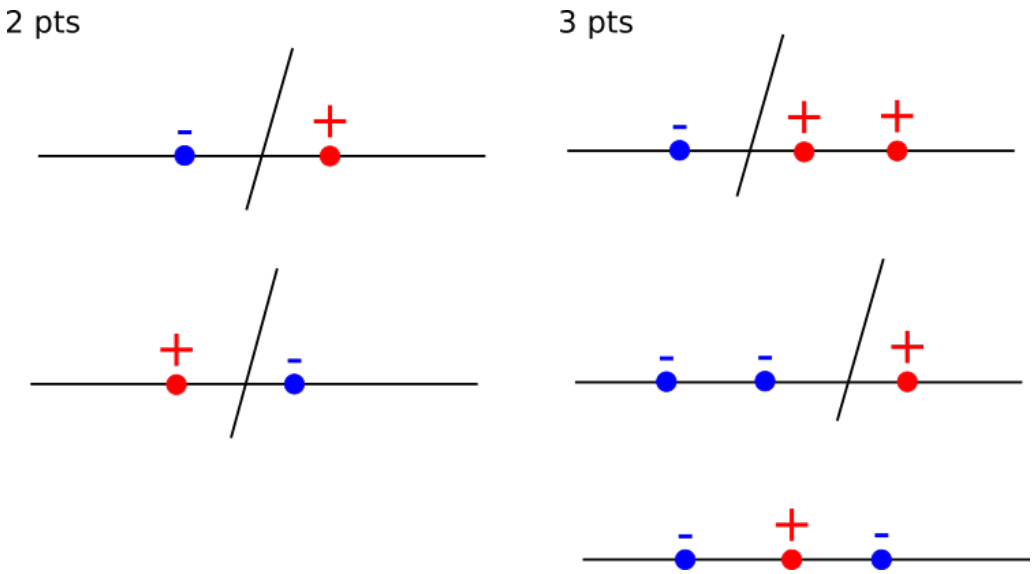

Figure: Any labeling is possible on 2 points, but not on 3

Shattering - 2-D Pictorial representation

If the function class $\mathcal{F}$ consists of linear classifiers only, then :

- How many can be shattered by linear classifiers in 2-dimensions ?

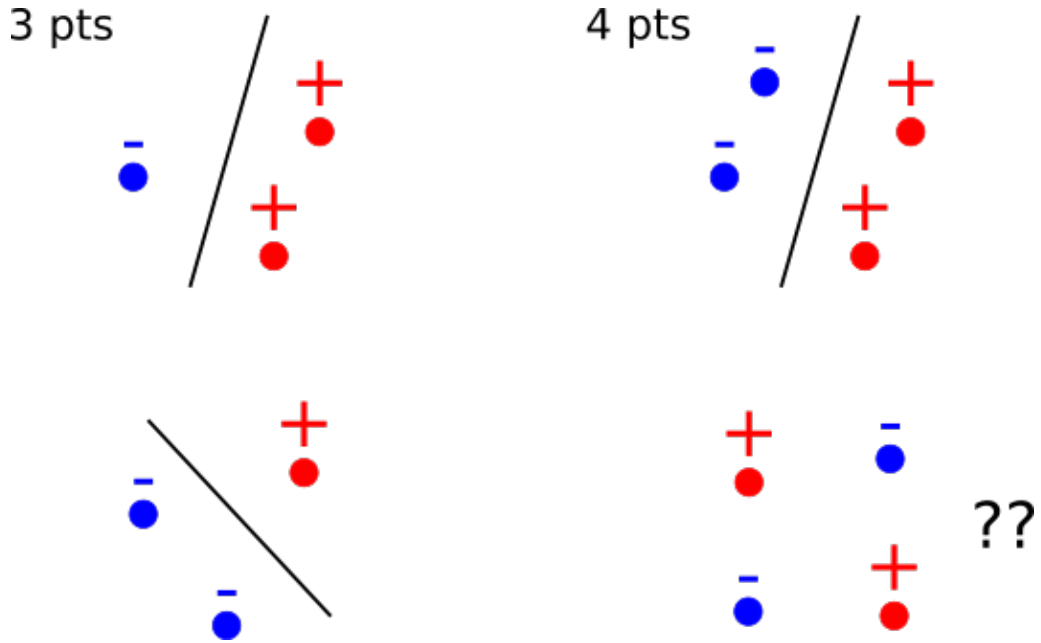

Figure: Any labeling is possible on 3 points, but not on 4

Shattering - 3-D Pictorial representation

If the function class $\mathcal{F}$ consists of linear classifiers only, then :

- How many points can be shattered by linear classifiers in 3-dimensions?

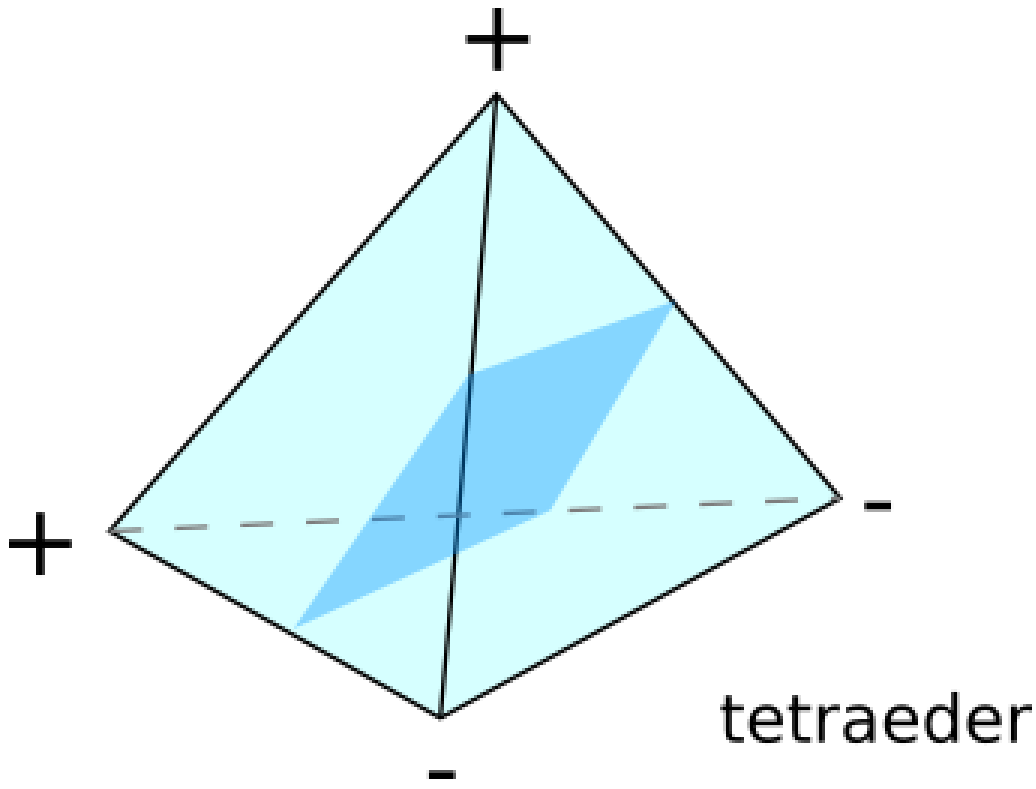

Figure: Any labeling is possible on 4 points, but not on 5

Uniform Convergence Bounds - I

We can now finally bound $P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon)$ in the following way :

$$
\begin{aligned}
& P \sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon) \\
& \leq 2P(\sup_{f \in \mathcal{F}} |R_{emp}(f) - R'_{emp}(f)| > \epsilon/2) \text{ (By symmetrization)} \\
& = 2P(\sup_{f \in \mathcal{F}_{Z_{2n}}} |R_{emp}(f) - R'_{emp}(f)| > \epsilon/2) \text{ (By Shattering argument)} \\
& \leq 2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4) \text{ (Union and Chernoff's bound)}
\end{aligned}
$$

Uniform Convergence Bounds - II

Key inequality

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon) \leq 2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4)$$

We consider two following important cases :

- When $\mathcal{N}(\mathcal{F}, 2n)$ grows polynomially with n, i.e., $\mathcal{N}(\mathcal{F}, 2n) \leq (2n)^k$ for some constant k

Uniform Convergence Bounds - II

Key inequality

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon) \leq 2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4)$$

We consider two following important cases :

- When $\mathcal{N}(\mathcal{F}, 2n)$ grows polynomially with n, i.e., $\mathcal{N}(\mathcal{F}, 2n) \leq (2n)^k$ for some constant k
  - $2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4) \leq 2 \times (2n)^k \times \exp(-n\epsilon^2/4) = 2\exp(k \log(2n) - n\epsilon^2/4)$

Uniform Convergence Bounds - II

Key inequality

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon) \leq 2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4)$$

We consider two following important cases :

- When $\mathcal{N}(\mathcal{F}, 2n)$ grows polynomially with n, i.e., $\mathcal{N}(\mathcal{F}, 2n) \leq (2n)^k$ for some constant k
  - $2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4) \leq 2 \times (2n)^k \times \exp(-n\epsilon^2/4) = 2\exp(k\log(2n) - n\epsilon^2/4)$
  - $2\exp(k\log(2n) - n\epsilon^2/4)$ goes to 0 as $n \to \infty$ satisfying the condition for Uniform Convergence, and hence implying consistency of ERM

Uniform Convergence Bounds - II

Key inequality

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon) \leq 2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4)$$

We consider two following important cases :

- When $\mathcal{N}(\mathcal{F}, 2n)$ grows polynomially with n, i.e., $\mathcal{N}(\mathcal{F}, 2n) \leq (2n)^k$ for some constant k
  - $2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4) \leq 2 \times (2n)^k \times \exp(-n\epsilon^2/4) = 2\exp(k\log(2n) - n\epsilon^2/4)$
  - $2\exp(k\log(2n) - n\epsilon^2/4)$ goes to 0 as $n \to \infty$ satisfying the condition for Uniform Convergence, and hence implying consistency of ERM
- Use the function class $\mathcal{F}_{all}$ (even consisting of highly non-smooth over-fitting functions), a class that can classify each sample in any way desired, $\mathcal{N}(\mathcal{F}, 2n) = 2^{2n}$
  - $2\mathcal{N}(\mathcal{F}, 2n) \exp(-n\epsilon^2/4) = 2 \times 2^{2n} \times \exp(-n\epsilon^2/4) = 2\exp(n(2\log(2) - \epsilon^2/4))$
  - Does not go to 0 as $n \to \infty$ implying inconsistency of ERM when the function class contains all functions

VC Dimension

Definition

VC dimension of a function class (denoted by $VC(\mathcal{F})$) is the maximum value n for which there exists a sample of size n that is shattered of $\mathcal{F}$.

Theorem

Empirical Risk minimization is consistent with respect to $\mathcal{F}$ if and only if VC-dimension of $\mathcal{F}$ is finite.

Other concepts

- Rademacher Complexity

Summary

- Abstract study of Supervised Learning
- Types of error
  - Empirical Error, Expected Error, Generalization gap
  - Estimation and Approximation error
- Consistency
  - When can ERM be inconsistent?
  - Uniform Convergence
  - Shattering

References

- Reference of Learning Theory material by Ulrike von Luxbourg
  - Statistical Learning Theory: Models, Concepts, and Results `https://arxiv.org/abs/0810.4752`
- Chapter 5 of the Learning with Kernels (Shoelkopf and Smola) book :
  - `http://agbs.kyb.tuebingen.mpg.de/lwk/`