CS:E4830 Kernel Methods in Machine Learning

Lecture 4 : Introductory Statistical Learning Theory

Rohit Babbar

30th January, 2019

LibShortText Solver

During the first part of the lecture, we covered the LibShortText[1] solver

- An example of text classification for short texts such as those obtained on e-commerce sites like eBay or Amazon
- To demonstrate the bigram features which are implicitly generated by polynomial kernel of degree two
- It is instructive to download and try this solver out with various settings and options provided
- The related paper is `https://www.csie.ntu.edu.tw/~cjlin/papers/libshorttext.pdf`

[1] `https://www.csie.ntu.edu.tw/~cjlin/libshorttext/`

Generalization in Machine Learning

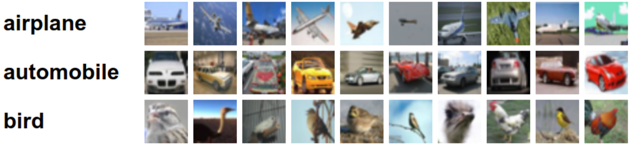

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

Generalization in Machine Learning

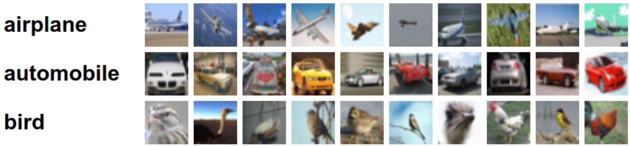

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

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? Make a good guess!

Generalization in Machine Learning

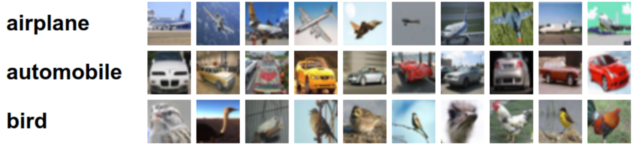

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

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? Make a good guess!
- Now, keep the training set images same but randomly shuffle their labels
  - Keep the test set same as the previous case
  - Train a deep net on the training set with randomized labels, and test on the test set,

Generalization in Machine Learning

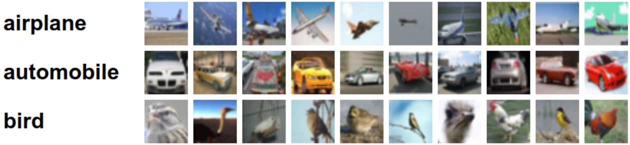

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

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? Make a good guess!
- Now, keep the training set images same but randomly shuffle their labels
  - Keep the test set same as the previous case
  - Train a deep net on the training set with randomized labels, and test on the test set,
  - What are the training and test errors?

Generalization in Machine Learning

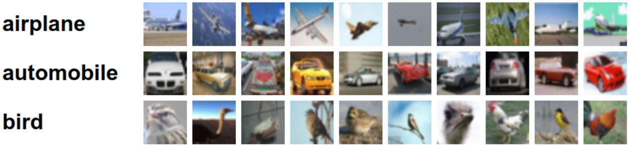

Figure: Some examples from three of the **ten classes** from CIFAR-10 dataset (others being **cat**, **deer**, **dog**, **frog**, **horse**, **ship**, **truck**) are shown above. Dataset contains 50,000 training and 10,000 test images for a total of 6,000 images per class

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? Make a good guess!
- Now, keep the training set images same but randomly shuffle their labels
  - Keep the test set same as the previous case
  - Train a deep net on the training set 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 ?

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 (generalization)

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
  - 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 - I

Assumptions

- Makes no assumption on the underlying data generating distribution P - (unlike in many cases where a distribution such as Gaussian is assumed and the goal is to find the parameters of that distribution)
- Labels can be noisy - $\eta(x) = P(y = 1|X = x)$. Below is an example of a two class problem, where joint distribution $P(x, C)$ is plotted for two classes C_1 and C_2 for a one dimensional input.

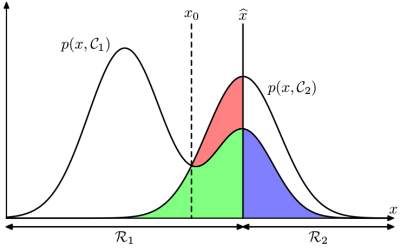

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

Assumptions of SLT - II

Assumptions

- Training points are sampled independently
  - The above assumption may not hold in certain practical situations (such as time series data), and hence require some other techniques
- The distribution P over $\mathcal{X} \times \mathcal{Y}$ is fixed and does not change w.r.t. time
- Distribution P which generates the data is unknown while learning
  - P is accessed indirectly through the training data

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 - (1)

Let's say $C_1 = +1$, and $C_2 = -1$ in the figure below. Also $P(.)$ in the text refers to the probablity and $p(.)$ in the picture refers to its density, but of the same object.

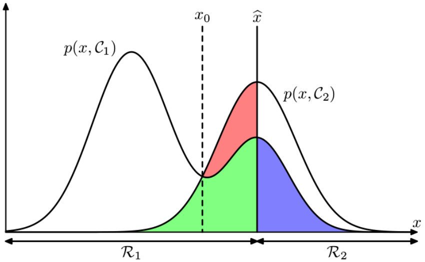

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

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

Bayes Classifier - (1)

Let's say $C_1 = +1$, and $C_2 = -1$ in the figure below. Also $P(.)$ in the text refers to the probablity and $p(.)$ in the picture refers to its density, but of the same object.

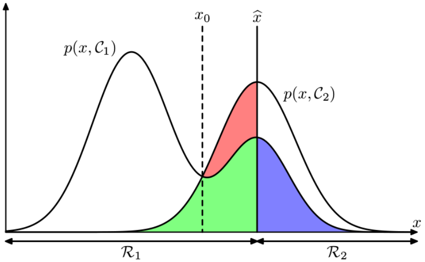

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

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

$$f_{Bayes}(x) := \begin{cases} C_1 & \text{if } P(y = C_1 | X = x) \geq 0.5 \\ C_2 & \text{otherwise} \end{cases}$$

Bayes Classifier - (2)

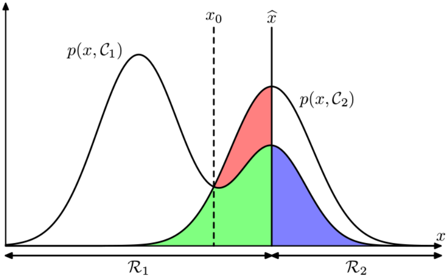

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

- Given a point, 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 - (2)

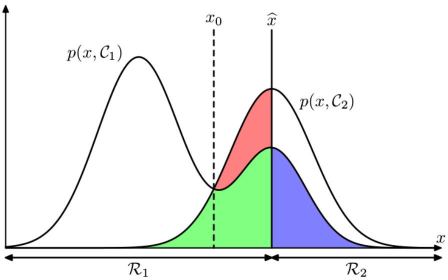

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

- Given a point, say $x = \hat{x}$, how do we compute $P(y = C_1|X = \hat{x})$ or $P(y = C_2|X = \hat{x})$?
- At what point in the graph $P(y = C_1|X = x) = 0.5$?

Bayes Classifier - (2)

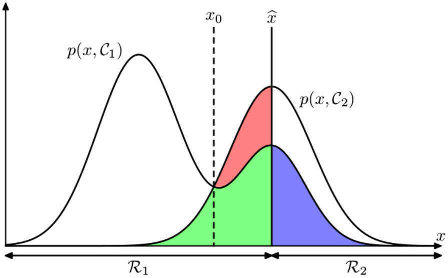

Figure: Depiction of noisy labels (picture from Chris Bishop's book)

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

Notion of Generalization

It is desired that the error of our classifier is close to that of Bayes classifier. However, another desirable quality in machine learning algorithms is **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 of f_n is low, i.e.,

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

- Note that having low generalization gap does imply low expected or test error, it just means that **empirical error is a good indicator of expected error**

Overfitting

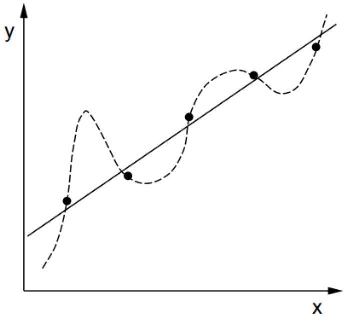

Figure: Overfitting example

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 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 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
- Lets call 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 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
- Lets call 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

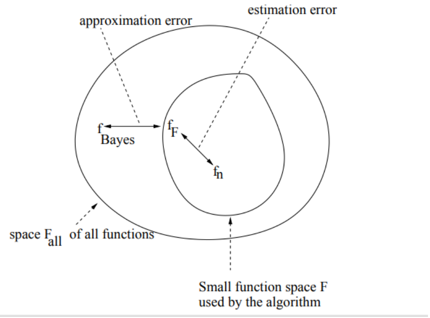

Figure: Pictorial depiction of the components of classification error

- 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

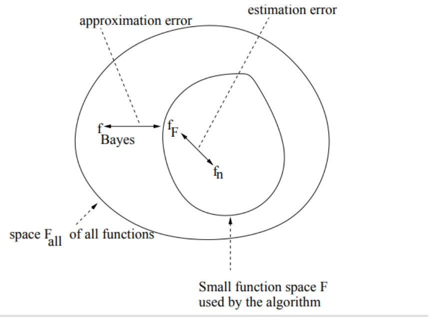

Figure: Pictorial depiction of the components of classification error

- 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 SVM, which of the two errors is actually going down?

Error variation with Function class capacity

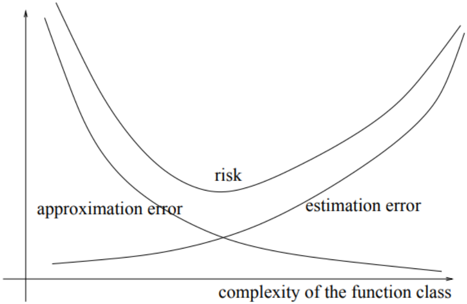

Figure: Variation of error components with the complexity of function class (tutorial by Von Luxburg and Schoelkopf)

- To the left with low complexity function class -
  - Linear classifiers or kernel classifier with high variance
- To the right with high complexity function class -
  - Deep neural networks

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 $\mathcal{F}$

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

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
- The motivation for the consistency of the principle of ERM comes from the law of large numbers, which we discuss next.

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 P) 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

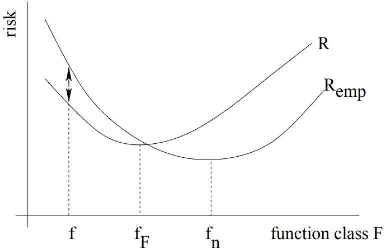

Figure: Depiction of training error and test error and various functions of interest

- 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

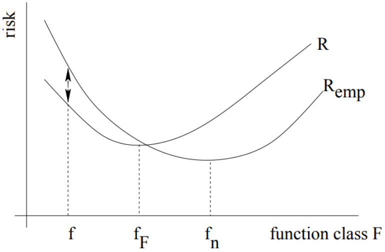

Figure: Depiction of training error and test error and various functions of interest

- 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 choose more and more training sets, by Chernoff's bound, for **every fixed function**, $R_{emp}(f)$ converges to $R(f)$,

Pictorial representation

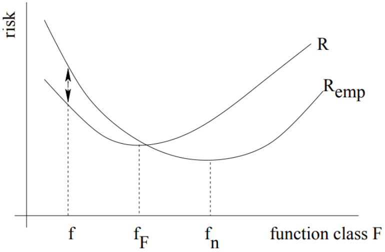

Figure: Depiction of training error and test error and various functions of interest

- 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 choose more and more training sets, by Chernoff's bound, for **every fixed function**, $R_{emp}(f)$ converges to $R(f)$,
- 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

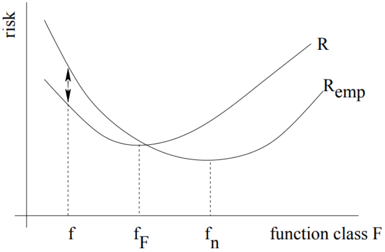

Figure: Depiction of training error and test error and various functions of interest

- 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 choose more and more training sets, by Chernoff's bound, for **every fixed function**, $R_{emp}(f)$ converges to $R(f)$,
- 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 $R(f_F)$

When can ERM be inconsistent?

An Empirical Risk Minimization Example

- Typically, in a machine learning setup, we do not have access to the true underlying data distribution, and instead we have access to a fixed training set $(x_i, y_i)_{i=1}^n$
- Assume that the data lies in $[0, 1]$, i.e., $x \in \mathcal{X} = [0, 1]$
  - Input x is chosen uniformly at random on $\mathcal{X}$,
  - the label y is chosen in a deterministic way as follows :

$$y = \begin{cases} -1 & \text{if } x < 0.5 \\ +1 & \text{otherwise} \end{cases}$$

- Consider, a potential classifier based on n training samples given as follows :

$$f_n(x) = \begin{cases} y_i & \text{if } x = x_i \text{ for some } i = 1 \dots n \\ +1 & \text{otherwise} \end{cases}$$

- What is it error on the training set?
  - training error $= 0$
  - Has it learnt anything?

When can ERM be incosistent?

An Empirical Risk Minimization Example

- What is it error on the training set?
    - training error $= 0$ (minimum possible)
    - Has it learnt anything?
- What is its test error?

When can ERM be incosistent?

An Empirical Risk Minimization Example

- What is it error on the training set?
  - training error $= 0$ (minimum possible)
  - Has it learnt anything?
- What is its test error?
- Therefore, the Empirical risk minimizer is not converging to the best function in the class
- Why does it happen?
  - Because we allow any function (could be highly non-smooth) in our function space
- In order to generalize, we need to **restrict our function class** by imposing some condition, which we study next.

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?

- Reference of Learning Theory material by Ulrike von Luxbourg
  - Statistical Learning Theory: Models, Concepts, and Results
    `https://arxiv.org/abs/0810.4752`
- Chris Bishop's book (available online) for Bayes classifier
  - Pattern Recognition and Machine Learning