Machine Learning I

Lecture 4: The Bias-Complexity Tradeoff

Nathaniel Bade

Northeastern University Department of Mathematics

Table of contents

Prior Knowledge

Prior Knowledge

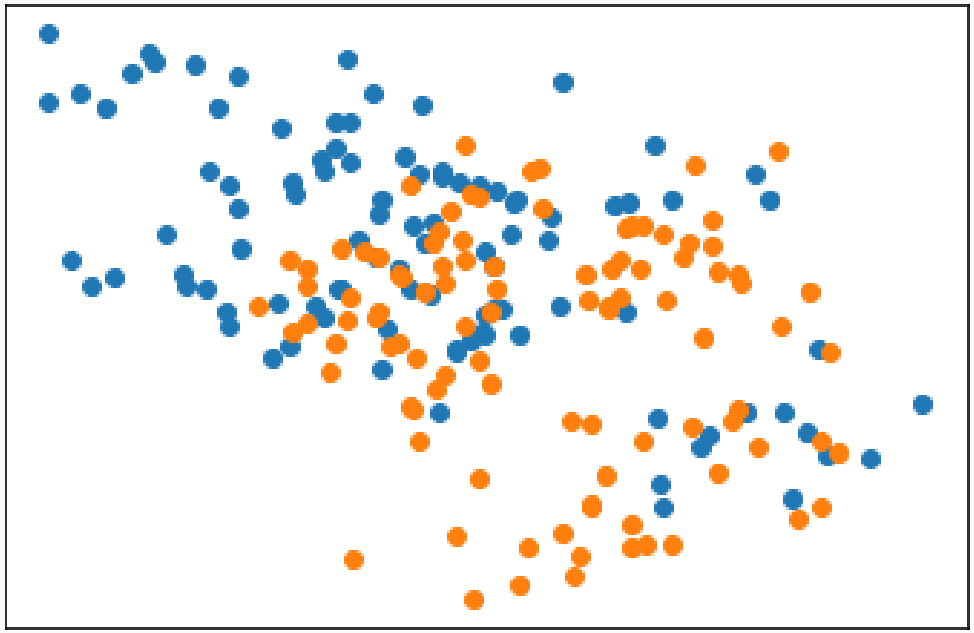
In the last few lectures we have seen that overfitting can mislead statistical learning, leading to a large difference between the empirical error and the true error.

We also saw that with sufficient data and a restricted hypothesis class we can effectively overcome the problem of over fitting. Precisely, with probability $1 - \delta$ the learner returns a classifier with true error within ϵ of the lowest true error of all classifiers in the hypothesis class.

For finite hypothesis classes, we showed that the sample complexity was bounded by

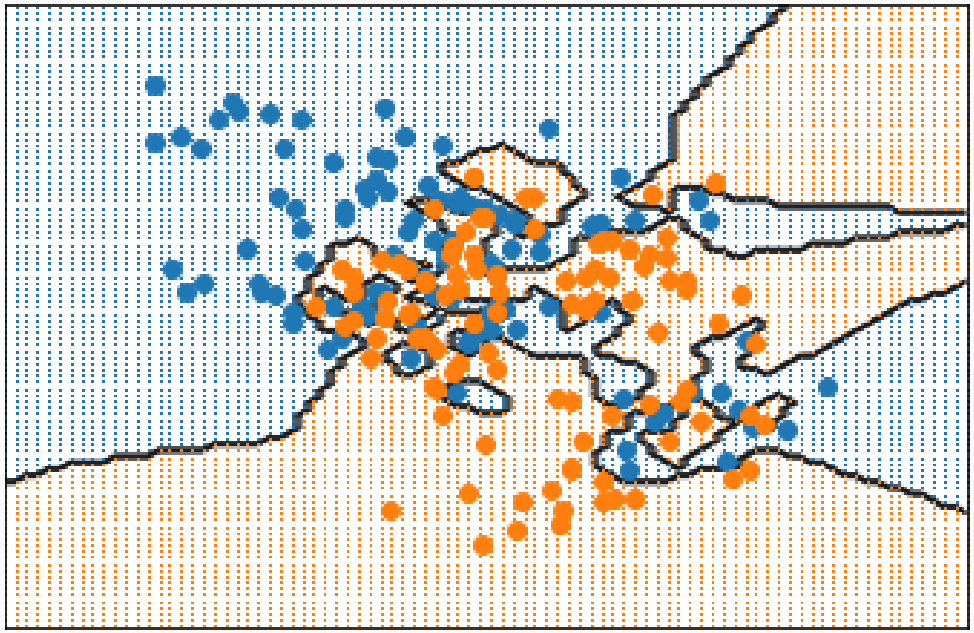
$$N_{\mathcal{H}}(\epsilon, \delta) \leq \text{ceil}\left(\frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2}\right).$$

Prior Knowledge



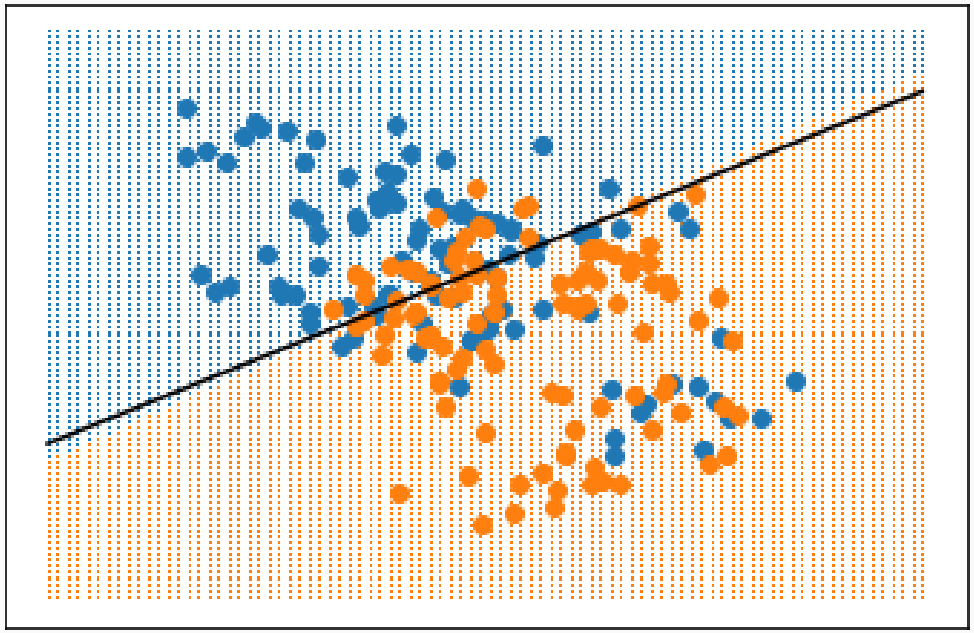
However, the choice of how to construct an appropriate hypothesis class seems to require an understanding of the problem. How was the data generated? Can we get a rough idea of the distribution on $\mathcal{X}$? A rough idea of the labeling scheme?

Prior Knowledge



However, the choice of how to construct an appropriate hypothesis class seems to require an understanding of the problem. How was the data generated? Can we get a rough idea of the distribution on $\mathcal{X}$? A rough idea of the labeling scheme?

Prior Knowledge



However, the choice of how to construct an appropriate hypothesis class seems to require an understanding of the problem. How was the data generated? Can we get a rough idea of the distribution on $\mathcal{X}$? A rough idea of the labeling scheme?

Prior Knowledge

All of the considerations above are known as **prior knowledge**, and they immediately raise another question: Is prior knowledge necessary? Is there a best algorithm for fitting data whose input is $\mathcal{X} = \mathbb{R}^n$, *regardless of the distribution*?

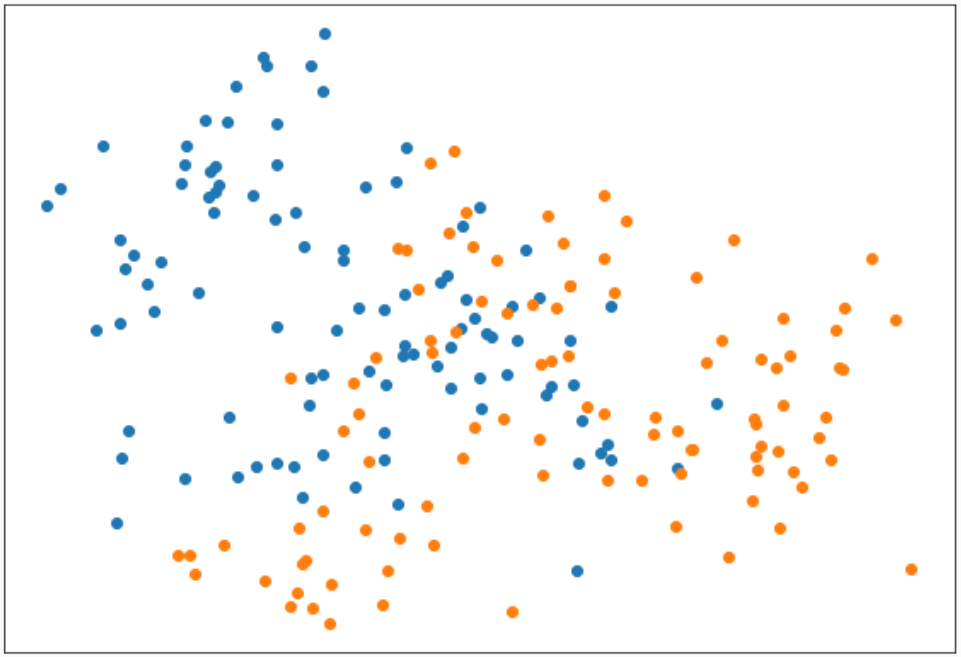
More precisely, a specific **learning task** is defined by an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. For a learning task, the goal of the learner is to find a predictor h whose risk is small with respect to $\mathcal{D}$.

Prior Knowledge

For example, classifying 100×100 pixel pictures as cats or dogs has the same sample space $\mathcal{X} \times \mathcal{Y} = R^{100 \times 100} \times \{0, 1\}$ as classifying 100×100 pixel MRI scans as containing cancerous cells or not. The only difference between the tasks are the distributions on $\mathcal{X} \times \mathcal{Y}$.

Does there exist a universal learner that can account for both cases? Is there an A and an N such that if A receives a training set of N i.i.d. samples it will (probably) return a predictor h with low risk, regardless of $\mathcal{D}$?

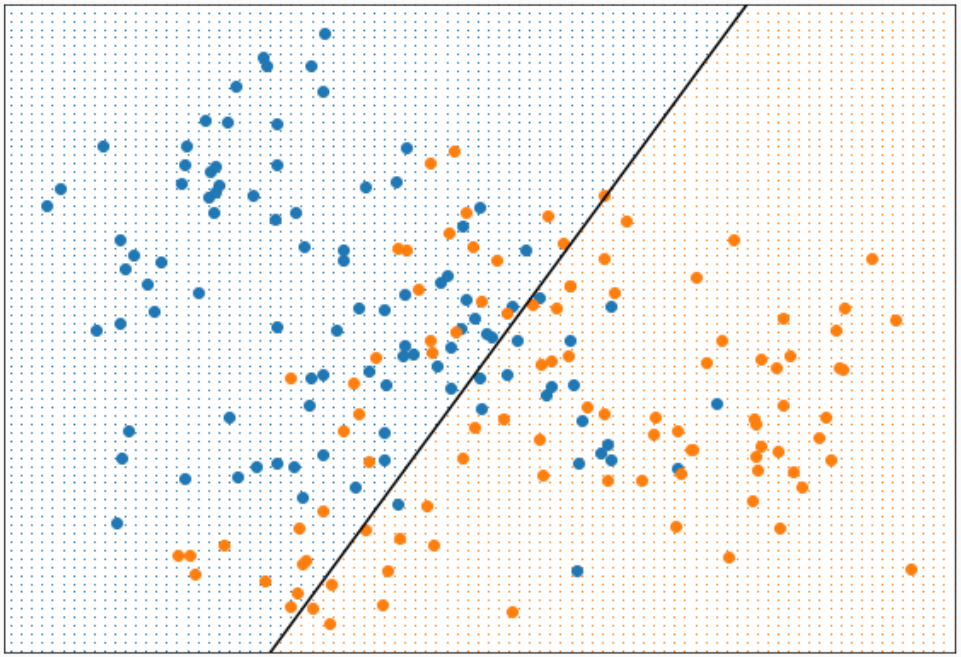
No such thing



The **no free lunch theorem** answers that question in the negative: For binary classification, every learner A has a distribution on which is fails.

We will say a learner fails if, upon receiving a training set $\mathcal{T}$ of i.i.d. data from a distribution, the learner is likely to have a large risk $L_{\mathcal{D}}(\, A(\mathcal{T})\,)$. There also must exists another learner that takes the same data and outputs a hypothesis class with small risk.

Failure of Learning



For example, a learner might do well given one distribution on $\mathcal{X} \times \mathcal{Y}$ but fail quite horribly on another. The challenge becomes cooking up a distribution on which a given learning algorithm fails.

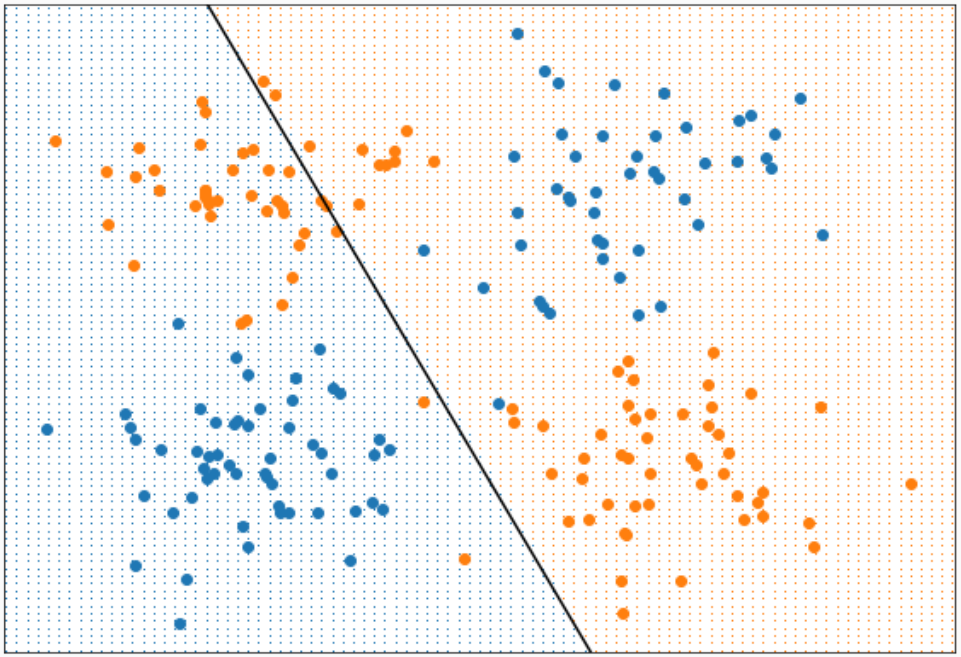
Question: Can you think of a learnable distribution on which the linear classifier fails?

Failure of Learning

For example, a learner might do well give one distribution on $\mathcal{X} \times \mathcal{Y}$ but fail quite horribly on another. The challenge becomes cooking up a distribution on which a given learning algorithm fails.

Question: Can you think of a learnable distribution on which the linear classifier fails?

Failure of Learning



For a learner to have "failed", there must exist an algorithm that can learn the distribution. There are certain distributions on which no learner can do well.

For example, it would not be a failure of the learner to not predict a truly random distribution, on which every point had an equal probability of returning each label.

Prior Knowledge

Practically, when you approach a problem its important to use any and all information you can to select a hypothesis class.

In the next section, we will prove the no free lunch theorem, and discuss some of it's implications.

In the second part of the lecture, we will discuss the decomposition of the total error into error due to the hypothesis class (**approximation error**) and the error due to overfitting the training data (**estimation error**). This decomposition of risk is known variously as the bias-complexity or bias-variance trade off.

No Free Lunch Theorem

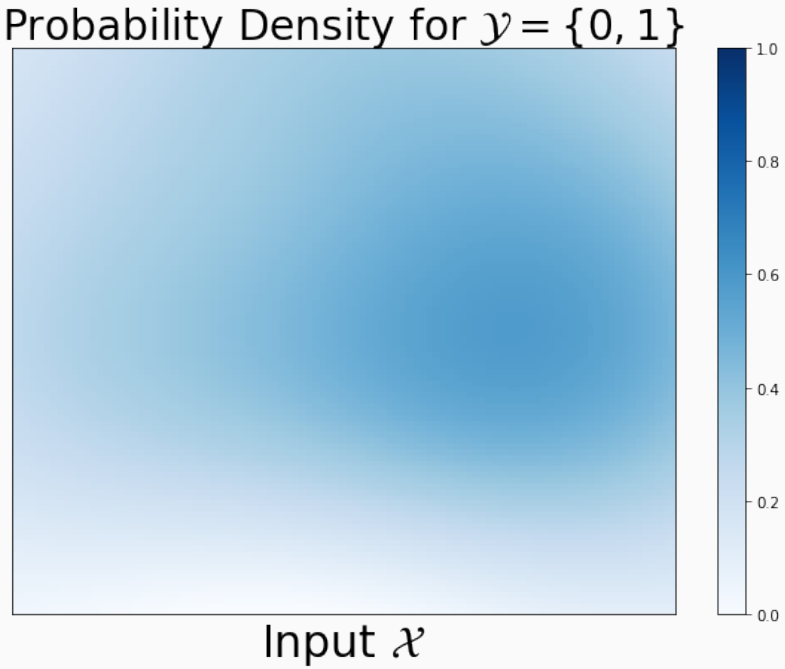
Statement

No Free Lunch Theorem: Let A be a learning algorithm for the task of binary classification with respect to 0-1 loss. Here, $\mathcal{X}$ may be arbitrary but $\mathcal{Y} = \{0, 1\}$. Let $N \leq |\mathcal{X}|/2$. Then, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times [0, 1]$ such that

1. There exists $f : \mathcal{X} \to [0, 1]$ with $L_{\mathcal{D}}(f) = 0$.
2. With probability $\delta > 1/7$, $L_{\mathcal{D}}(A(T)) \geq 1/8$.

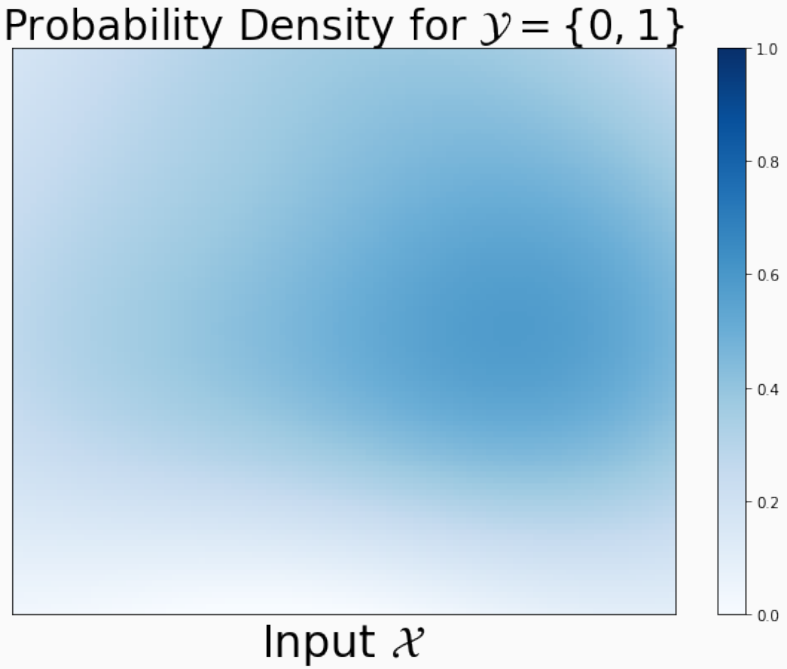
Comments: In words, every learner A on $\mathcal{X} \times \mathcal{Y}$ has a learnable task (ie a distribution) on which it fails. The condition $N \leq |\mathcal{X}|/2$ is technical but intuitive, if a learner has access to all the data we would expect the theorem not to hold.

Ideas in Proof



Lets give a high level overview before we run through the computations. There are only a few key ideas but they are good to be fluent in.

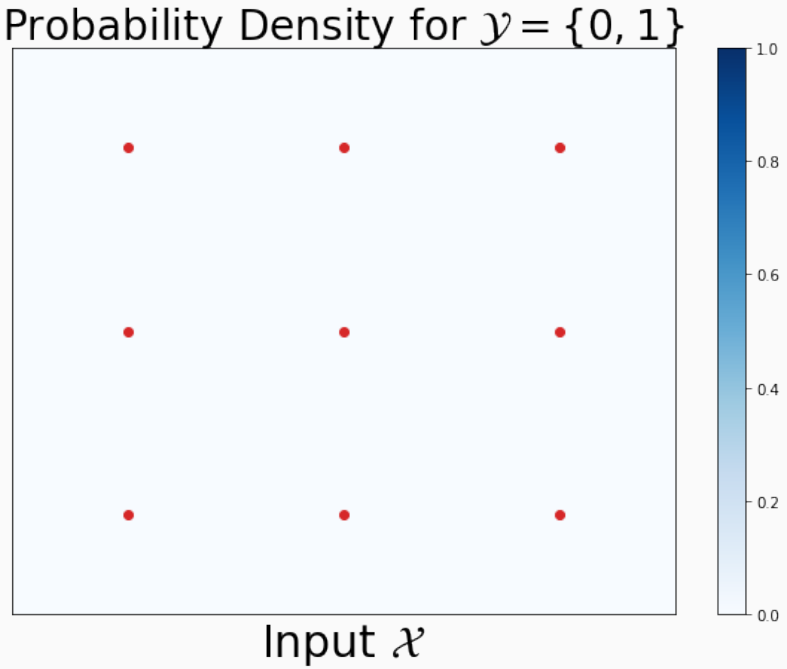
Ideas in Proof



First, we fix N since the bound is (mostly) independent training size. We don't need to consider all distributions, instead we are trying to produce a distribution on which a learning algorithm A fails.

As such, it is enough to localize to a finite set $C \subseteq \mathcal{X}$, since any distribution constructed there can be extended to a distribution on all of $\mathcal{X}$ by thickening.

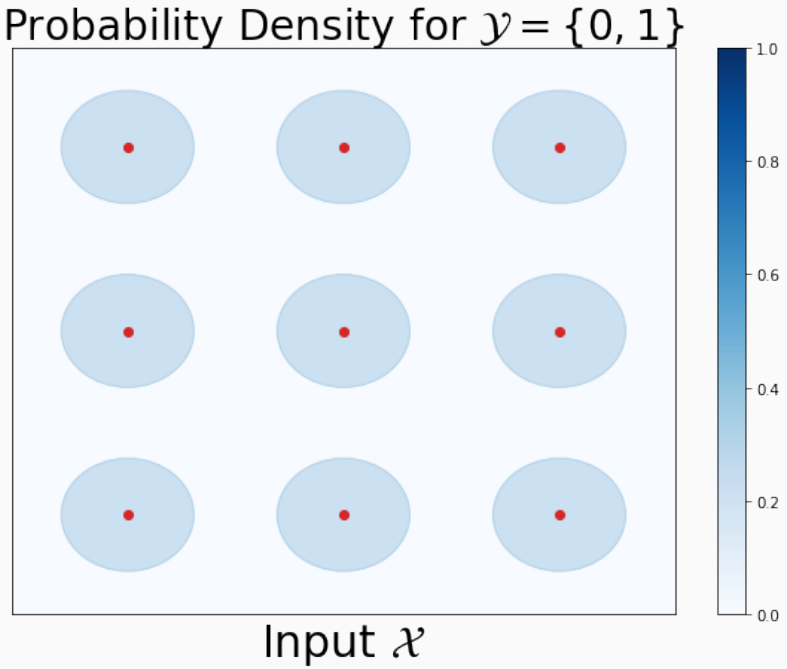
Ideas in Proof



First, we fix N since the bound is (mostly) independent training size. We don't need to consider all distributions, instead we are trying to produce a distribution on which a learning algorithm A fails.

As such, it is enough to localize to a finite set $C \subseteq \mathcal{X}$, since any distribution constructed there can be extended to a distribution on all of $\mathcal{X}$ by thickening.

Ideas in Proof



First, we fix N since the bound is (mostly) independent training size. We don't need to consider all distributions, instead we are trying to produce a distribution on which a learning algorithm A fails.

As such, it is enough to localize to a finite set $C \subseteq \mathcal{X}$, since any distribution constructed there can be extended to a distribution on all of $\mathcal{X}$ by thickening.

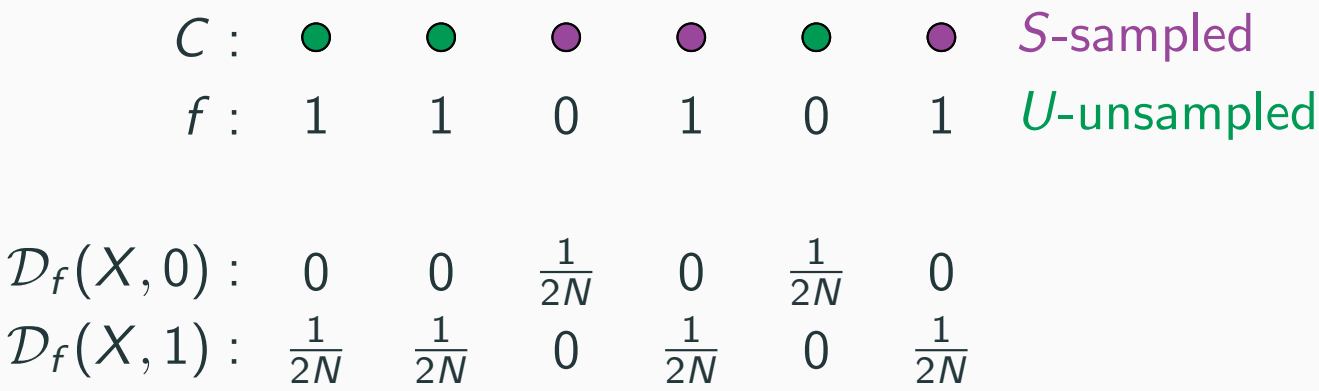
Ideas in Proof

C :	●	●	●	●	●	●
f :	1	1	0	1	0	1
$\mathcal{D}_f(X,0)$:	0	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0
$\mathcal{D}_f(X,1)$:	$\frac{1}{2N}$	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$

We will work instead over labelings of the discrete set $C \subset \mathcal{X}$, where $|C| = 2N$. The space of all labelings $\mathbb{L}$ is finite and contains 2^{2N} labels f. For each labeling $f \in \mathbb{L}$, we define the following distribution, uniform over $\mathcal{X}$:

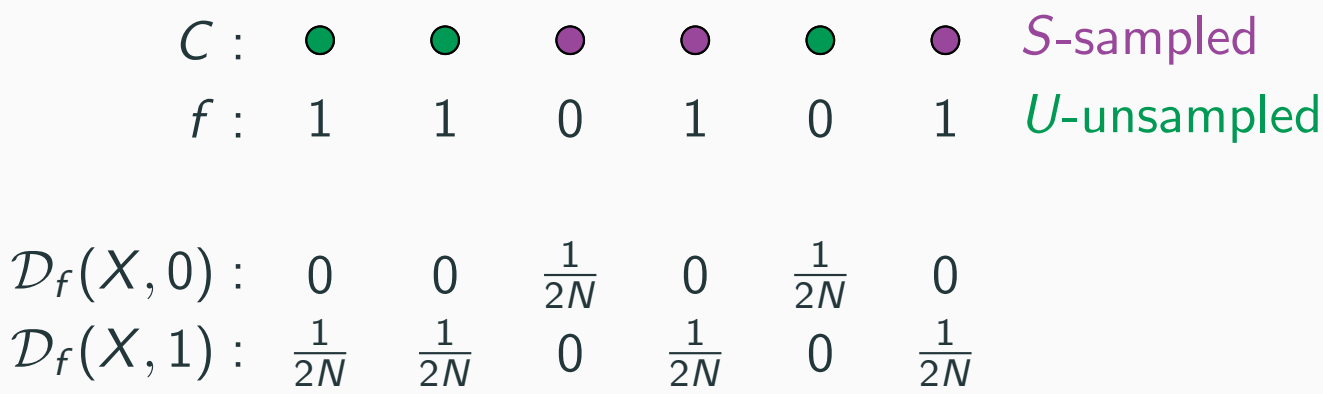
$$\mathcal{D}_f((X,Y)) = \begin{cases} \frac{1}{2N} & \text{if } f(X) = Y\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Ideas in Proof



Let $\mathbb{T}$ be the set of all size N samples of the C. The cardinality of this set is $|\mathbb{T}| = (2N)^N$, and points may be sampled more than once. For any sampled set $S \in \mathbb{T}$, let $U = C - S$ be the unsampled points.

Ideas in Proof



Effectively what we have done is a Cartesian decomposition of the space training set with all labels into two pieces: The space of all labels $\mathbb{L}$ and the space of all sample sets $\mathbb{T}$.A training set is then denoted

$$S_f = \{(x, f(x)) : x \in S\}.$$

The set $\mathbb{L} \times \mathbb{T}$ is the set of all training sets with any labeling.

Ideas in Proof

C :	●	●	●	●	●	●	S-sampled
f :	1	1	0	1	0	1	U-unsampled
$\mathcal{D}_f(X,0)$:	0	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	
$\mathcal{D}_f(X,1)$:	$\frac{1}{2N}$	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	

Idea: Let $A(S_f)$ denote the classifier returned by the training set

$$S_f = \{(x, f(x)) : x \in S\}.$$

We will show that

$$\max_{f \in \mathbb{L}} E_S\left[L_{D_f}(A(S_f))\right] \geq \frac{1}{4}.$$

That is, we will show that the maximum (over all labelings) of the expected error (for the classifier returned by A trained on S_f) is bounded from below.

Ideas in Proof

C :	●	●	●	●	●	●	S-sampled
f :	1	1	0	1	0	1	U-unsampled
$\mathcal{D}_f(X,0)$:	0	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	
$\mathcal{D}_f(X,1)$:	$\frac{1}{2N}$	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	

This 1/4 bound has a simple interpretation: Since we may draw the same point more than once, $|S| \leq 2N$ and $|U| \geq 2N$. That means that any algorithm isn't going to know how to label the (possibly more than) $\frac{1}{2}$ of the data in U.

Furthermore, we would imagine that on average the algorithm will mislabel approximately 50% of the labels. So no matter what the algorithm is it should mislabel around 1/4 of the points.

Min/Max Swap

We will use the following trick one and a half times:

Let $\mathcal{D}$ be a joint distribution on $\mathcal{X} \times \mathcal{Y}$ and let $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ be a positive function. We can bound the maximum of the expectation value with respect to one variable by the minimum expectation value with respect to the other:

$$\begin{aligned} \max_X E_Y\big[g(X,Y)\big] &\geq E_X E_Y\big[g(X,Y)\big] \\ &\geq E_Y E_X\big[g(X,Y)\big] && \text{by Fubini's Theorem} \\ &\geq \min_Y E_X\big[g(X,Y)\big] \end{aligned}$$

This might seem very trivial, but we will use it swap between expectations over the labels $\mathbb{L}$ and over the sample sets $\mathbb{T}$.

Proof of No Free Lunch Theorem

Min/Max Swap

$C:$							$\mathcal{T}$-training
$f:$	1	1	0	1	0	1	U-unsampled
							$\mathbb{T}$ Sample Space
$\mathcal{D}_f(X,0):$	0	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\mathbb{L}$ Label Space
$\mathcal{D}_f(X,1):$	$\frac{1}{2N}$	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	

Proof of NFL: Let C, $\mathbb{T}$ and $\mathbb{L}$ be as above. We can use the max/min swap to write

$$\max_{f\in\mathbb{L}} E_S\left[L_{D_f}(A(S_f))\right] \geq \min_{S\in\mathbb{T}} E_{f\in\mathbb{L}}\left[L_{D_f}(A(S_f))\right].$$

We will now attempt to bound $L_{D_f}(A(S_f))$.

Min/Max Swap

$C:$							$\mathcal{T}$-training
$f:$	1	1	0	1	0	1	U-unsampled
							$\mathbb{T}$ Sample Space
$\mathcal{D}_f(X,0):$	0	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\mathbb{L}$ Label Space
$\mathcal{D}_f(X,1):$	$\frac{1}{2N}$	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	

For fixed sample S,

$$L_{D_f}(A(S_f)) = \frac{1}{2N}\sum_{x\in C} 1_{f(x)\neq A(S_f)[x]}$$

is the number of incorrect predictions divided by $2N$. We expect the error on the sampled set to be low, so we bound by the error on U

$$L_{D_f}(A(S_f)) \geq \frac{1}{2N}\sum_{x\in U} 1_{f(x)\neq A(S_f)[x]} \geq \frac{1}{2|U|}\sum_{x\in U} 1_{f(x)\neq A(S_f)[x]}.$$

Min/Max Swap

Writing then

$$L_{D_f}(A(Sf)) \geq \frac{1}{2} E_{x \in U}\left[1_{f(x) \neq A(S_f)[x]} \right],$$

we can use the fact the min is at most the expectation value to write

$$\begin{aligned} E_{f \in \mathbb{L}}\left[L_{\mathcal{D}_f}(A(S_f)) \right] &\geq \frac{1}{2} E_{f \in \mathbb{L}} E_{x \in U}\left[1_{f(x) \neq A(S_f)[x]} \right] \\ &\geq \frac{1}{2} \min_{x \in U} E_{f \in \mathbb{L}}\left[1_{f(x) \neq A(S_f)[x]} \right]. \end{aligned}$$

So for a fixed S, the average error (over all labelings) is bounded by the $1/2$ the average error over the worst element in the unsampled set.

Min/Max Swap

For every function $f \in \mathbb{L}$, there is a function $f' \in \mathbb{L}$ such that f and f' agree on the sample set $f|_S = f'|_S$, but assign opposite labels to the unsampled set, $f(x) \neq f'(x)$, $\forall x \in U$. For every such a pair

$$1_{f(x) \neq A(S_f)[x]} + 1_{f'(x) \neq A(S_f)[x]} = 1\,, \ \forall x \in U\,.$$

But then

$$E_{f \in \mathbb{L}}\left[\, 1_{f(x) \neq A(S_f)[x]} \,\right] = \frac{1}{|\mathbb{L}|} \sum_{f \in \mathbb{L}} 1_{f(x) \neq A(S_f)[x]} = \frac{1}{2}\,.$$

Lets put it all together.

Min/Max Swap

Lets put it all together.

$$\begin{aligned}
\max_{f \in \mathbb{L}} E_S \left[L_{D_f}(A(S_f)) \right] &\geq \min_{S \in \mathbb{T}} E_{f \in \mathbb{L}} \left[L_{D_f}(A(S_f)) \right] \\
&\geq \min_{S \in \mathbb{T}} \min_{x \in U} \frac{1}{2} E_{f \in \mathbb{L}} \left[1_{f(x) \neq A(S_f)[x]} \right] \\
&\geq \min_{S \in \mathbb{T}} \min_{x \in U} \frac{1}{2} \cdot \frac{1}{2} \\
&= \frac{1}{4}.
\end{aligned}$$

We have shown that there is always a labeling on which the risk of an algorithm A is bounded from below. The final step is to show that this result isn't rare.

Min/Max Swap

$C:$							$\mathcal{T}$-training
$f:$	1	1	0	1	0	1	U-unsampled
							$\mathbb{T}$ Sample Space
$\mathcal{D}_f(X,0):$	0	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\mathbb{L}$ Label Space
$\mathcal{D}_f(X,1):$	$\frac{1}{2N}$	$\frac{1}{2N}$	0	$\frac{1}{2N}$	0	$\frac{1}{2N}$	

Exercise: Use Markov's Inequality to show that

$$\max_{f \in \mathbb{L}} E_S \left[L_{D_f}(A(S_f)) \right] \geq \frac{1}{4}$$

implies

$$\mathbb{P}\left[L_{\mathcal{D}}(A(S)) \geq 1/8 \right] \geq 1/7 \,.$$

This final step completes the proof.

Question: What sort of distribution must k-NN fail on?

The Bias Complexity Tradeoff

Avoiding Underfitting

The no free lunch theorem isn't the end of machine learning, it simply asserts that there is no universally best learner for every task. If fact, it implies that we should use any prior knowledge to avoid learners that perform poorly on a distribution. Such prior knowledge can be expressed by restricting the hypothesis class.

But how do we choose this class? On one hand, we want a class that contains a classifier that will return the minimum error. On the other hand, the class of all functions is clearly not learnable so we cant just choose the richest class.

Bias and Complexity

To begin to address the question, we decompose the total error

$$L_{\mathcal{D}}(h_{\mathcal{T}}) = \epsilon_{app} + \epsilon_{est} = \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \left(L_{\mathcal{D}}(h_{\mathcal{T}}) - \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)\right)$$

into

Approximation Error: The minimum risk achievable within a class. This term is also known as the **inductive bias**.

Estimation Error: The difference between the minimum achievable risk and the ERM error. This is the term to be minimized in APAC.

Estimation Error

Last lecture we derived an expression for the estimation error of a finite hypothesis class $\mathcal{H}$. We found,

$$\epsilon_{est} = L_{\mathcal{D}}(h_{\mathcal{T}}) - \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \geq \left(\frac{\log(2|\mathcal{H}|/\delta)}{2N} \right) .$$

For fixed δ, if **estimation error** increases, the training size N must decrease or $|\mathcal{H}|$ must increase (exponentially). This justifies thinking of ϵ_{est} as a measure of relative complexity.

The Bias Variance Tradeoff: Bias-complexity at a point

Estimation Error

There is more we can say if we restrict ourselves to a specific loss functions and distributions. Assume for the moment that there is a true labeling function $f : \mathcal{X} \to \mathcal{Y}$. Consider the pointwise **mean squared error** of a predictor $h_{\mathcal{T}}$ on some $x_0 \in \mathcal{X}$.

$$MSE(x_0) = E_{\mathcal{T}}[f(x_0) - h_{\mathcal{T}}(x_0)]^2$$

Define $y_0^{\mathcal{H}} := E_{\mathcal{T}}[h_{\mathcal{T}}(x_0)]$ to be the expected label of x_0 over the training sets. Then

$$\begin{aligned} MSE(x_0) &= E_{\mathcal{T}}\Big[\big(f(x_0) - y_0^{\mathcal{H}}\big) + \big(y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0)\big) \Big]^2 \\ &= E_{\mathcal{T}}\big[f(x_0) - y_0^{\mathcal{H}} \big]^2 + E_{\mathcal{T}}\big[y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0) \big]^2 \\ &\quad + 2E_{\mathcal{T}}\big[\big(f(x_0) - y_0^{\mathcal{H}}\big)\big(y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0)\big) \big] \end{aligned}$$

Estimation Error

Lets analyze the cross term in

$$MSE(x_0) = E_{\mathcal{T}}\left[f(x_0) - y_0^{\mathcal{H}} \right]^2 + E_{\mathcal{T}}\left[y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0) \right]^2 + 2E_{\mathcal{T}}\left[\left(f(x_0) - y_0^{\mathcal{H}}\right)\left(y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0)\right)\right].$$

Since $f(x_0) - y_0^{\mathcal{H}}$ is constant, and $y_0^{\mathcal{H}} - E_{\mathcal{T}}[h_{\mathcal{T}}(x_0)] = y_0^{\mathcal{H}} - y_0^{\mathcal{H}} = 0$, the cross term vanishes

$$\begin{aligned} E_{\mathcal{T}}\left[\left(f(x_0) - y_0^{\mathcal{H}}\right)\left(y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0)\right)\right] \\ &= \left(f(x_0) - y_0^{\mathcal{H}}\right) E_{\mathcal{T}}\left(y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0)\right) \\ &= 0\,. \end{aligned}$$

So the mean squared error decomposes into a sum of squared terms

$$MSE(x_0) = E_{\mathcal{T}}\left[f(x_0) - y_0^{\mathcal{H}} \right]^2 + E_{\mathcal{T}}\left[y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0) \right]^2.$$

Estimation Error

Bias-Variance decomposition:

$$MSE(x_0) = E_{\mathcal{T}}\left[f(x_0) - y_0^{\mathcal{H}} \right]^2 + \left[y_0^{\mathcal{H}} - h_{\mathcal{T}}(x_0) \right]^2.$$

Pointwise,

> **Variance**: $\mathrm{Var}_{\mathcal{T}}(h_{\mathcal{T}}(x_0)) = E_{\mathcal{T}}\left[E_{\mathcal{T}}[h_{\mathcal{T}}(x_0)] - h_{\mathcal{T}}(x_0) \right]^2$ is the sampling variance of the estimate of y_0.
>
> **Bias**: $\mathrm{Bias}^2(h_{\mathcal{T}}(x_0)) = \left(f(x_0) - y_0^{\mathcal{H}} \right)^2$ here is the difference between the average labeling and the true labeling.

Notice that our two definitions of bias match up and that the variance accounts for the estimation error.

References

This lecture covers chapter 5 of Shalevl-Shwarts and Ben-David and section 2.5 of Hastie, Tibshirani and Friedman.