Course outline

1. Introduction, Feasibility of learning, Bias - Variance tradeoff (*May 16*)
2. Linear regression, Logistic regression (*May 18*)
3. Regularization, Validation (*May 23*)
4. K-means, Principal Component Analysis (*May 25*)
5. Laboratory

- Theory
- Technique
- Practical

Machine Learning: Lecture 1

Introduction

Feasibility of learning

Bias - Variance tradeoff

Mirko Mazzoleni

16 May 2017

University of Bergamo
Department of Management, Information and Production Engineering
mirko.mazzoleni@unibg.it

Resources

These lectures give only a glimpse of the vast machine learning field. Additional material (not required for the exam) can be found using the following *free* resources

MOOCs

- Learning from data (*Yaser S. Abu-Mostafa - EDX*)
- Machine learning (*Andrew Ng - Coursera*)
- The analytics edge *(Dimitris Bertsimas - EDX)*
- Statistical learning *(Trevor Hastie and Robert Tibshirani - Standford Lagunita)*

Books

- An Introduction to Statistical Learning, with application in R *(Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani)*
- Neural Networks and Deep Learning *(Michael Nielsen)*

Outline

- Introduction
- Components of learning
- Puzzle
- Feasibility of learning
- Bias - variance tradeoff

Outline

- **Introduction**
- Components of learning
- Puzzle
- Feasibility of learning
- Bias - variance tradeoff

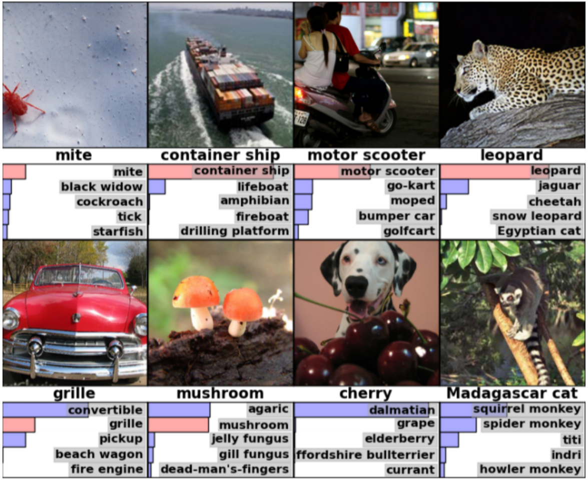
Why

Machine learning and data science have been deemed as the sexiest jobs of the 21th century [1]

- Virtually every aspect of business is now open to data collection
- Collected information need to be analyzed properly in order to get actionable results
- A huge amount of data requires specific infrastructures to be handled
- A huge amount of data requires computational power to be analyzed
- We can let computers to perform decisions given previous examples
- Rising of specific job titles
- . . . Fun ☺

Learning examples

Recent years: stunning breakthroughs in computer vision applications [2]



Learning examples

- Spam e-mail detection system
- Credit approval
- Recognize objects in images
- Find the relation between house prices and house sizes
- Identify the risk factors for prostate cancer

- Market segmentation
- Market basket analysis
- Language models (word2vec)
- Social network analysis
- Movies recommendation
- Low-order data representations

What learning is about

Machine learning is meaningful to be applied if:

1. A pattern exist
2. We cannot pin it down mathematically
3. We have data on it

Assumption 1. and 2. are not mandatory:

- If a pattern does not exist, I do not learn anything
- If I can describe the mathematical relation, I will not presumably learn the best function
- The real constraint is assumption 3

Outline

- Introduction
- **Components of learning**
- Puzzle
- Feasibility of learning
- Bias - variance tradeoff

Components of learning

Formalization:

- Input: $\mathbf{x}$ (*e-mail textual content*) $\rightarrow$ each dimension is some e-mail attribute
- Ouptut: y (*spam/not spam?*) $\rightarrow$ the decision that we have to take in the end
- Target function: $f : \mathcal{X} \rightarrow \mathcal{Y}$ (*Ideal spam filter formula*) $\rightarrow$ unknown, we have to learn it
- Data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$ (*historical records of e-mail examples*)

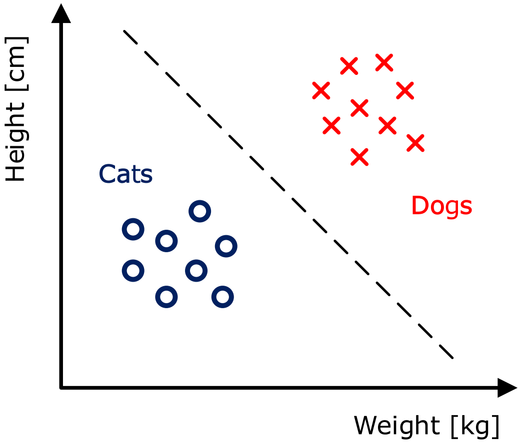
$\downarrow \quad \downarrow \quad \downarrow$

- Hypothesis: $g : \mathcal{X} \rightarrow \mathcal{Y},\ g \in \mathcal{H}$ (*formula to be used*) $\rightarrow$ g is an approximation of f

$\mathcal{H}$ is called the Hypothesis space. This, toghether with the Learning algorithm, form the *learning model*

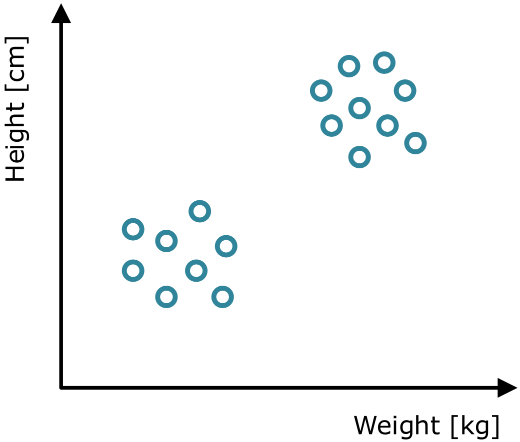
Supervised learning

- The "correct answer" y is given
- Predict y from a set of inputs $\mathbf{x} \in \mathbb{R}^{d \times 1}$
- Regression: predict continous output $y \in \mathbb{R}$ (real value)
- Classification: predict discrete output $y \in \{1, \ldots, C\}$ (class)



Unsupervised learning

- Instead of (**input, correct output**) we get (**input, ?**)
- Find properties of the inputs $\mathbf{x} \in \mathbb{R}^{d \times 1}$
- High-level representation of the input
- Elements into the same cluster have similar properties



Learning examples revisited

Supervised Learning (Classification)

- Spam e-mail detection system
- Credit approval
- Recognize objects in images
- Find the relation between house prices and house sizes
- Identify the risk factors for prostate cancer

Unsupervised Learning

- Market segmentation
- Market basket analysis
- Language models (word2vec)
- Social network analysis
- Movies recommendation*
- Low-order data representations

Learning examples revisited

Supervised Learning (Regression)

- Spam e-mail detection system
- Credit approval
- Recognize objects in images
- Find the relation between house prices and house sizes
- Identify the risk factors for prostate cancer

Unsupervised Learning

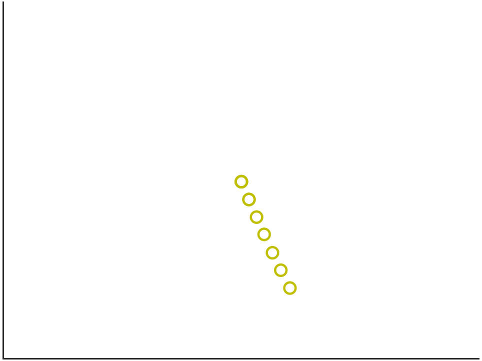
- Market segmentation
- Market basket analysis
- Language models (word2vec)
- Social network analysis
- Movies recommendation*
- Low-order data representations

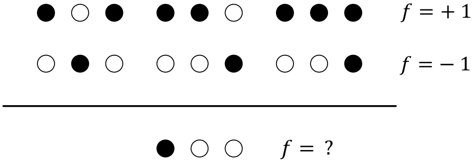
Outline

- Introduction
- Components of learning
- **Puzzle**
- Feasibility of learning
- Bias - variance tradeoff

Puzzle

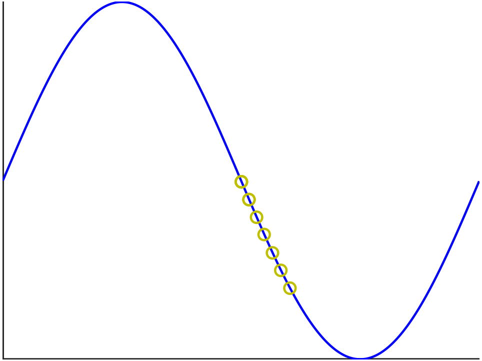
Which are the plausible response values of the unknown function, on positions of the input space that we have not seen?

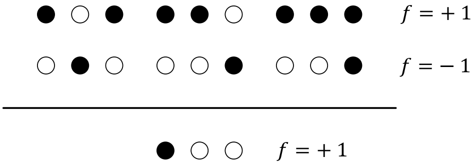




Puzzle

It is not possible to know how the function behaves outside the observed points (*Hume's induction problem* [3])





If first dot is black $\rightarrow f = +1$

Outline

- Introduction
- Components of learning
- Puzzle
- **Feasibility of learning**
- Bias - variance tradeoff

Feasibility of learning

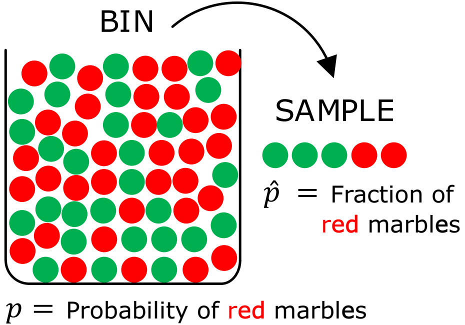
Focus on supervised learning, dichotomic classification case

Problem: Learning an unknown function

Solution: Impossible ☹. The function can assume any value outside the data we have

Experiment

- Consider a 'bin' with red and green marbles
- $\mathbb{P}[$ picking a red marble $] = p$
- The value of p is unknown to us
- Pick N marbles independently
- Fraction of red marbles in the sample $= \hat{p}$



Does $\hat{p}$ say something about p?

No!

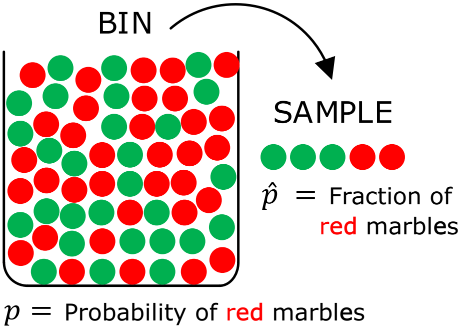
Sample can be mostly green while bin is mostly red

Possible

Yes!

Sample frequency $\hat{p}$ is likely close to bin frequency p (*if the sample is sufficiently large*)

Probable



What does $\hat{p}$ says about p?

In a big sample (large N), $\hat{p}$ is probably close to p (within ε)

This is stated by the **Hoeffding's inequality:**

$$\mathbb{P}[|\hat{p} - p| > \varepsilon] \leq 2e^{-2\varepsilon^2 N}$$

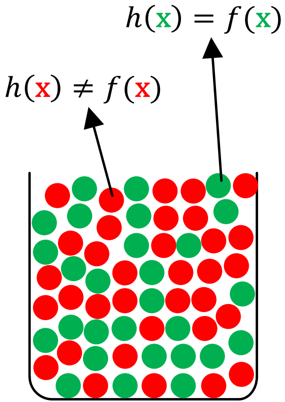
The statement $p = \hat{p}$ is P.A.C. (Probably Approximately Correct)

- The quantity $|\hat{p} - p| > \varepsilon$ is a bad event, we want its probability to be low
- The bound is valid for all N and $\varepsilon \rightarrow$ ε is a margin of error
- The bound does not depend on p
- If we set for a lower margin ε, we have to increase the data N in order to have a small probability of the bad event happening

Connection to learning

Bin: The unknown is a number p

Learning: The unknown is a function $f : \mathcal{X} \to \mathcal{Y}$



Each marble • is a input point $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$. For a specific hypothesis $h \in \mathcal{H}$:

- Hypothesis got it right $h(\mathbf{x}) = f(\mathbf{x})$
- Hypothesis got it wrong $h(\mathbf{x}) \neq f(\mathbf{x})$

Both p and $\hat{p}$ depend on the particular hypothesis h

$\hat{p} \to$ In sample error $E_{\text{in}}(h)$

$p \to$ Out of sample error $E_{\text{out}}(h)$

The **Out of sample error $E_{\text{out}}(h)$** is the quantity that really matters

There is a probability $P(\mathbf{x})$ of having observed the sampled data

Error measures

What does $h \approx f$ mean? Define an error measure: $E(h, f)$. Almost always *pointwise definition*: $e(h(\mathbf{x}), f(\mathbf{x}))$

Pointwise error examples

- *Squared error*: $e(h(\mathbf{x}), f(\mathbf{x})) = (h(\mathbf{x}) - f(\mathbf{x}))^2$
- *Binary error*: $e(h(\mathbf{x}), f(\mathbf{x})) = \mathbb{I}[h(\mathbf{x}) \neq f(\mathbf{x})]$

The error measure should be specified by the user

It is interesting to look at the *overall error*:

Overall error examples

- *In sample error*: $E_{\text{in}} = \frac{1}{N} \sum_{n=1}^{N} e(h(\mathbf{x}_n), f(\mathbf{x}_n))$
- *Out of sample error*: $E_{\text{out}} = \mathbb{E}_{\mathbf{x}}[e(h(\mathbf{x}), f(\mathbf{x}))]$

Connection to *real* learning

In a learning scenario, the function h is not fixed a priori

- The *learning algorithm* is used to fathom the hypothesis space $\mathcal{H}$, to find the best hypothesis $h \in \mathcal{H}$ that matches the sampled data $\rightarrow$ call this hypothesis g
- The Hoeffding's inequality does not hold for multiple hypothesis
- With many hypotheses, there is more chance to find a good hypothesis g only by chance $\rightarrow$ the function can be perfect on sampled data but bad on unseen ones

The Hoeffding's inequality becomes:

$$\mathbb{P}[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \varepsilon] \leq 2Me^{-2\varepsilon^2 N}$$

where M is the number of hypotheses in $\mathcal{H}$ $\rightarrow$ M can be infinity ☹

The quantity $E_{\text{out}}(g) - E_{\text{in}}(g)$ is called the generalization error

It turns out that the number of hypotheses M can be replaced by a quantity (called the growth function) which is eventually bounded by a polynomial

- This is due to the fact the the M hypotheses will be very overlapping $\to$ they generate the same "classification dichotomy"
- The growth function $m_{\mathcal{H}}(N)$ indicates the maximum number of dichotomies that can be generated by a function $h \in \mathcal{H}$ on a finite set of N points $\to$ the maximum possible number of dichotomies that can be generated on N points is 2^N

By replacing M with $m_{\mathcal{H}}(N)$, it is possible to obtain the following result

Vapnik-Chervonenkis Inequality

$$\mathbb{P}[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \varepsilon] \leq 4 m_{\mathcal{H}}(2N) e^{-\frac{1}{8}\varepsilon^2 N}$$

Generalization theory

The **VC-dimension** is a single parameter that characterizes the growth function

Definition

The Vapnik-Chervonenkis dimension of a hypothesis set $\mathcal{H}$*, denoted by* $d_{\mathsf{vc}}(\mathcal{H})$ *or simply* d_{vc}*, is the largest value of* N *for which* $m_{\mathcal{H}}(N) = 2^N$*. If* $m_{\mathcal{H}}(N) = 2^N$ *for all* N*, then* $d_{\mathsf{vc}}(\mathcal{H}) = \infty$

It can be shown that:

- If the d_{vc} is finite, than $m_{\mathcal{H}} \leq N^{d_{\mathsf{vc}}} + 1 \rightarrow$ this is a polynomial that will eventually be dominated by $e^{-N} \rightarrow$ generalization guarantees
- For linear models $y = \sum_{i=1}^{d} \alpha_i \mathrm{x}_i + \beta$, $d_{\mathsf{vc}} = d + 1 \rightarrow$ can be interpreted as the number of effective parameters

Rearranging things

Start from the VC inequality:

$$\mathbb{P}[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \varepsilon] \leq \underbrace{4m_{\mathcal{H}}(2N)e^{-\frac{1}{8}\varepsilon^2 N}}_{\delta}$$

Get ε in terms of δ:

$$\delta = 4m_{\mathcal{H}}(2N)e^{-\frac{1}{8}\varepsilon^2 N} \implies \varepsilon = \underbrace{\sqrt{\frac{8}{N}\ln\frac{4m_{\mathcal{H}}(2N)}{\delta}}}_{\Omega}$$

Interpretation

- I want to be at most $\varepsilon\%$ away from E_{out}, given that I have E_{in}
- I want this statement to be correct $(1-\delta)\%$ of the times
- Given any two of N, δ, ε, it is possible to compute the remaining element

Generalization bound

Following previous reasoning, it is possible to say that, with probability $1-\delta$:

$$|E_{\text{in}}(g) - E_{\text{out}}(g)| \leq \Omega(N, \mathcal{H}, \delta) \implies -\Omega(N, \mathcal{H}, \delta) \leq E_{\text{in}}(g) - E_{\text{out}}(g) \leq \Omega(N, \mathcal{H}, \delta)$$

Solving for the inequalities leads to:

1. $E_{\text{out}}(g) \geq E_{\text{in}}(g) - \Omega(N, \mathcal{H}, \delta)$ → not of much interest ☹
2. $\boldsymbol{E_{\text{out}}(g) \leq E_{\text{in}}(g) + \Omega(N, \mathcal{H}, \delta)}$ → bound on the out of sample error! ☺

Observations

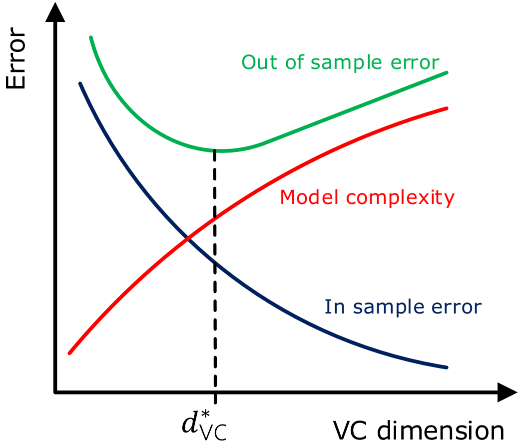
- $E_{\text{in}}(g)$ is known
- The penalty Ω can be computed if $d_{\text{vc}}(\mathcal{H})$ is known

Generalization bound

Analysis of the generalization bound $E_{\text{out}}(g) \leq E_{\text{in}}(g) + \Omega(N, \mathcal{H}, \delta)$

- $\Omega \uparrow$ if $d_{\text{vc}} \uparrow \rightarrow$ penalty for model complexity
- $\Omega \uparrow$ if $\delta \uparrow \rightarrow$ penalty for higher confidence
- $\Omega \downarrow$ if $N \uparrow \rightarrow$ less penalty with more examples
- $E_{\text{in}} \downarrow$ if $d_{\text{vc}} \uparrow \rightarrow$ a more complex model can fit the data better

The optimal model is a compromise between E_{in} and Ω



Take home lessons

Rule of thumb

How many data points N are required to ensure a good generalization bound?

$$\boxed{N \geq 10 \cdot d_{\mathrm{vc}}}$$

General principle

Match the **'model complexity'** to the **data resources**, not to the **target complexity**

Outline

- Introduction
- Components of learning
- Puzzle
- Feasibility of learning
- **Bias - variance tradeoff**

Approximation vs. generalization

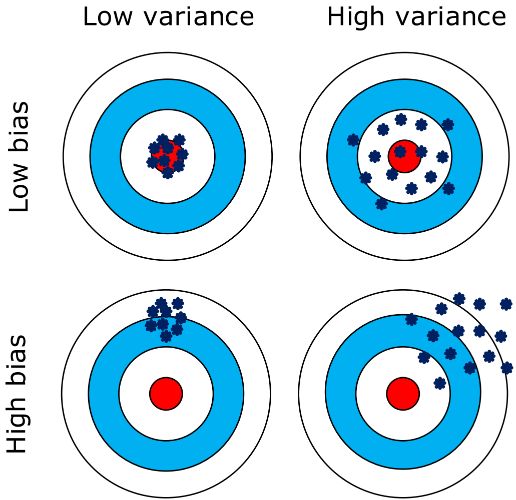
The ultimate goal is to have a small E_{out}: good approximation of f out of sample

- More complex $\mathcal{H} \Longrightarrow$ better chance of **approximating** f $\rightarrow$if $\mathcal{H}$ is too simple, we fail to approximate f and we end up with large E_{in}
- Less complex $\mathcal{H} \Longrightarrow$ better chance of **generalizing** out of sample $\rightarrow$if $\mathcal{H}$ is too complex, we we fail to generalize well because of the large model complexity term

VC analysis (discussed for binary classification) was one approach: $E_{\text{out}} \leq E_{\text{in}} + \Omega$

Bias-variance decomposition is another: it applies to **real valued targets** and uses **squared error** $\rightarrow$ the learning algorithm is not obliged to minimize squared error loss. However, we measure its produced hypothesis's bias and variance using squared error

Bias and variance



Bias and variance

Bias-variance analysis decomposes E_{out} into two terms:

1. How well $\mathcal{H}$ can approximate f $\rightarrow$ **Bias**
2. How well we can zoom in on a good $h \in \mathcal{H}$ $\rightarrow$ **Variance**

The out of sample error is (making explicit the dependence of g on $\mathcal{D}$):

$$E_{\text{out}}(g^{(\mathcal{D})}) = \mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

The expected out of sample error of the learning model is independent of the particular realization of data set used to find $g^{(\mathcal{D})}$:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right] &= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right] \\
&= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right]
\end{aligned}$$

Bias and variance

Focus on $\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$

Define the 'average' hypothesis $\bar{g}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}(\mathbf{x})\right]$

This average hypothesis can be derived by imagining many datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$, and building it by $\bar{g}(\mathbf{x}) \approx \frac{1}{N}\sum_{k=1}^{K} g^{(\mathcal{D}_k)}(\mathbf{x})$ $\rightarrow$ this is a conceptual tool, and $\bar{g}$ does not need to belong to the hypothesis set

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] &= \mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}) + \bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D}}\Big[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2 + \left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2 \\
&\quad + 2 \cdot \left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)\Big]
\end{aligned}$$

Bias and variance

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\textbf{var}(\mathbf{x})} + \underbrace{\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2}_{\textbf{bias}(\mathbf{x})}$$

Therefore:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[E_{\text{out}}(g^{(\mathcal{D})})\right] &= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]\right] \\
&= \mathbb{E}_{\mathbf{x}}\left[\textbf{bias}(\mathbf{x}) + \textbf{var}(\mathbf{x})\right] \\
&= \quad \textbf{bias} \quad + \quad \textbf{var}
\end{aligned}$$

Bias and variance

Interpretation

- The **bias** term $\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2$ measures how much our learning model is biased away from the target function

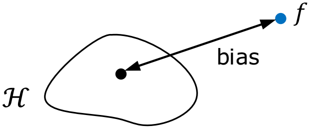
  Infact, $\bar{g}$ has the benefit of learning from an unlimited number of datasets, so it is only limited in its ability to approximate f by the limitations of the learning model itself

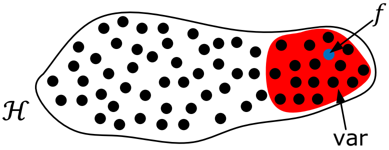
- The **variance** term $\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]$ measures the variance in the final hypothesis, depending on the data set, and can be thought as how much the final chosen hypothesis differs from the 'mean' (best) hypothesis

Bias and variance

$$\textbf{bias} = \left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2$$

$$\textbf{variance} = \mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]$$





Very small model. Since there is only one hypothesis, both the average function $\bar{g}$ and the final hypothesis $g^{(\mathcal{D})}$ will be the same, for any dataset. Thus, $\mathrm{var} = 0$. The bias will depend solely on how well this single hypothesis approximates the target f, and unless we are extremely lucky, we expect a large bias

Very large model. The target function is in $\mathcal{H}$. Different data sets will led to different hypotheses that agree with f on the data set, and are spread around f in the red region. Thus, $\mathrm{bias} \approx 0$ because $\bar{g}$ is likely to be close to f. The var is large (heuristically represented by the size of the red region)

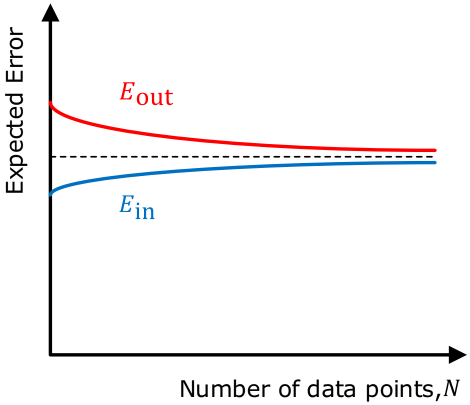
Learning curves

How it is possible to know if a model is suffering from bias or variance problems?

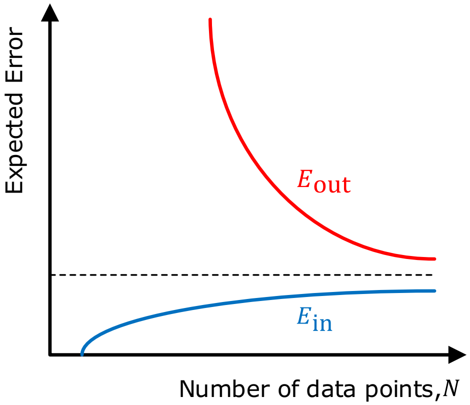
The learning curves provide a graphical representation for assessing this, by plotting the *expected out of sample error* $\mathbb{E}_{\mathcal{D}}[E_{\text{out}}(g^{\mathcal{D}})]$ and the *expected in sample error* $\mathbb{E}_{\mathcal{D}}[E_{\text{in}}(g^{\mathcal{D}})]$ vs. the number of data N

In the practice, the curves are computed from one dataset, or by dividing it into more parts and taking the mean curve resulting from various datasets

Learning curves



Simple model



Complex model

Learning curves

Interpretation

- Bias can be present when the error is quite high and E_{in} is similar to E_{out}
- When bias is present, getting more data is not likely to help
- Variance can be present when there is a gap between E_{in} and E_{out}
- When variance is present, getting more data is likely to help

Fixing bias

- Try adding more features
- Try polynomial feature
- Try a more complex model
- Boosting

Fixing variance

- Try a smaller set of features
- Get more training examples
- Regularization
- Bagging

References

[1] https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century Last accessed: 7 May 2017

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems, 2012

[3] Domingos, Pedro. The Master Algorithm. Penguin Books, 2016.