**ABOUT MACHINE LEARNING: (1 slide)**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), one aims to construct algorithms that are able to *learn* to predict a certain target output. To achieve this, the learning algorithm is presented some training examples that demonstrate the intended relation of input and output values. Then the learner is supposed to approximate the correct output, even for examples that have not been shown during training. Without any additional assumptions, this problem cannot be solved since unseen situations might have an arbitrary output value. The kind of necessary assumptions about the nature of the target function are subsumed in the phrase inductive bias.

**WHAT IS INDUCTIVE BIAS: (2 slides)**

The inductive bias (also known as learning bias) of a learning algorithm is the set of assumptions that the learner uses to predict outputs of given inputs that it has not encountered.

Approaches to a more formal definition of inductive bias are based on [mathematical logic](https://en.wikipedia.org/wiki/Mathematical_logic). Here, the inductive bias is a logical formula that, together with the training data, logically entails the hypothesis generated by the learner. However, this strict formalism fails in many practical cases, where the inductive bias can only be given as a rough description (e.g. in the case of [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_networks)).

**EXAMPLE:**

Example: In our example of a bit string classifier, assume that the learner has been given the strings { 1100, 1010 } as positive examples of some class of strings.

The set of all strings beginning with “1” and ending with “0,” the set of all strings beginning with “1,” the set of all strings of even parity , or any other subset of the entire population that includes { 1100, 1010 }. What criteria can the learner use to choose one of these generalizations ?

The data alone are not sufficient, all of these choices are consistent with that data. The learner must make additional assumptions about the nature of likely concepts. In learning, these assumptions often take the form of heuristics of choosing a branch of the search space.

**COMMON INDUCTIVE BIASES IN MACHINE LEARNING: (2-3 slides)**

* Maximum [conditional independence](https://en.wikipedia.org/wiki/Conditional_independence): if the hypothesis can be cast in a [Bayesian](https://en.wikipedia.org/wiki/Bayesian_inference) framework, try to maximize conditional independence. This is the bias used in the [Naive Bayes classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier).
* Minimum [cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)) error: when trying to choose among hypotheses, select the hypothesis with the lowest cross-validation error. Although cross-validation may seem to be free of bias, the ["no free lunch"](https://en.wikipedia.org/wiki/No_free_lunch_in_search_and_optimization) theorems show that cross-validation must be biased.
* Maximum margin: when drawing a boundary between two classes, attempt to maximize the width of the boundary. This is the bias used in [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machines). The assumption is that distinct classes tend to be separated by wide boundaries.
* [Minimum description length](https://en.wikipedia.org/wiki/Minimum_description_length): when forming a hypothesis, attempt to minimize the length of the description of the hypothesis. The assumption is that simpler hypotheses are more likely to be true. See [Occam's razor](https://en.wikipedia.org/wiki/Occam%27s_razor).
* Minimum features: unless there is good evidence that a [feature](https://en.wikipedia.org/wiki/Feature_space) is useful, it should be deleted. This is the assumption behind [feature selection](https://en.wikipedia.org/wiki/Feature_selection) algorithms.
* Nearest neighbours: assume that most of the cases in a small neighbourhood in [feature space](https://en.wikipedia.org/wiki/Feature_space) belong to the same class. Given a case for which the class is unknown, guess that it belongs to the same class as the majority in its immediate neighbourhood. This is the bias used in the [k-nearest neighbours algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm). The assumption is that cases that are near each other tend to belong to the same class.

**NEED OF BIAS: (1 slide)**

Without an Inductive Bias we have no rationale to choose one hypothesis over another and thus a random guess would be as good as any other option. Inductive Bias guides which hypothesis we should prefer? What happens in this case if we use simplicity (Occam’s Razor) as our inductive Bias?

**WHICH BIAS IS BEST: (1 slide)**

Not one Bias that is best on all problems

Experiments

– Over 50 real world problems

– Over 400 inductive biases

– mostly variations on critical variable biases vs. similarity biases

Different biases were a better fit for different problems

**LEARNABLE PROBLEMS: (1 SLIDE)**

“Raster Screen” Problem

Pattern Theory

– Regularity in a task

– Compressibility

Don’t care features and Impossible states

Interesting/Learnable Problems

– What we actually deal with

– Can we formally characterize them?

Learning a training set vs. generalizing

– A function where each output is set randomly (coin-flip)

– Output class is independent of all other instances in the data set