DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8

Machine Learning at DIT

INTRODUCTION TO INDUCTIVE LEARNING

*** *SOLUTIONS* ***

DUBLIN INSTITUTE OF TECHNOLOGY

Dr. John Kelleher and Dr. Brian Mac Namee

*** *SOLUTIONS* ***

*** *SOLUTIONS* ***

1. Distinguish between **supervised** and **unsupervised** learning.

The distinction is that with **supervised learning** we know the actual label or category for each piece of data on which we train, whereas with **unsupervised learning** we do not know the classification of the data in the training sample. Unsupervised learning can thus often be viewed as a **clustering** task, while supervised learning can usually be seen as a **classification** task, or equivalently as a function-fitting task where one extrapolates the shape of a function based on some data points.

2. Describe the differences between **lazy learners** and **eager learners**, giving examples of each.

Definitions:

Lazy learners do not try to build a model from the training data, but simply use it at classification time

Eager learners build a mode from the training data during training, and use only this model at classification time, ignoring the original data.

Key differences:

- Lazy methods may consider query instance when deciding how to generalise beyond the training data D; eager methods cannot since they have already chosen global approximation when seeing the query.
- **Efficiency** lazy learners require less training times but more time at prediction; eager learners require more training times by less time for prediction
- **Accuracy** lazy learners effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function; eager learners must commit to a single hypothesis that covers the entire instance space.
- It is easier for lazy learners to handle **concept drift**

Examples:

Lazy learning example : case based reasoning

Eager learning example : Decision=tree, neural networks, support vector machines

3. In the context of machine learning, explain what is meant by **overfitting** the training data.

   Overfitting occurs when classifiers make decisions based on accidental properties of the training set that will lead to errors on the test set (or new data). As a result, whenever there is a large set of possible hypotheses, one has to be careful not to use the resulting freedom to find meaningless ”regularity” in the data.

4. Explain what is meant by the term **abductive reasoning**.

   Abductive reasoning allows the antecedent (head) of a rule to concluded with the conclusion is true provided that doing so is consistent. Abductive reasoning is primarily diagnostic; given the effect find the likely cause.

5. Explain what is meant by **inductive learning**.

   Inductive Learning involves the process of learning by example where a system tries to induce a general rule from a set of observed instances

6. Explain what is meant by **reinforcement learning**.

   Rather than being told what to do by a teacher (cf. supervised learning) a reinforcement learning agent must learn from the reinforcement/reward each action recieves from the environment.

7. In the context of inductive learning explain what is meant by a **consistent hypothesis**.

   A hypothesis is consistent if it agrees with the true function on all examples that we have.

8. For some data sets it is possible to devise multiple hypotheses that are consistent with the data. Describe a heuristic for choosing among multiple consistent hypotheses and explain why your heuristic is reasonable.

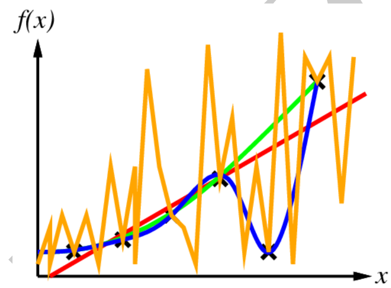
   One answer is to use Occams razor (sometimes called Ockhams razor) : prefer the hypothesis that maximizes a combination of simplicity and consistency with the data. This makes sense, because hypotheses that are no simpler than the data themselves are failing to extract any pattern from the data. Defining simplicity is not easy but it seems reasonable to say that a degree-1 polynomial is simpler than a degree-12 polynomial.

9. Distinguish between **classification learning** and **regression learning**.

- learning a discrete-valued function is called classification learning
- learning a continuous function is called regression.

10. Inductive machine learning is often referred to as an *ill-posed problem*. What is meant by this description?

Inductive machine learning algorithms essentially search through a hypothesis space to find a the best hypothesis that is consistent with the training data used. It is possible to find multiple hypotheses that are consistent with a given training set (i.e. agrees with all training examples). It is for this reason that inductive machine learning is referred to as an ill-posed problem as there is typically not enough information in the training data used to build a model to choose a single best hypothesis. Inductive machine learning algorithms must somehow choose one of the available hypotheses as the *best*. An example like that shown in the figure below would be useful at this point



11. Explain how do machine learning algorithms deal with the fact that machine learning is ill posed.

Because inductive learning is ill-posed, we have to make some extra assumptions to have a unique solution with the data we have. The set of assumptions we make to have learning possible is called the **inductive bias** of the learning algorithm - this is the main implication of inductive machine learning being ill-posed.

12. In the context of machine learning, explain what is meant by the term **inductive bias** and illustrate your explaination using examples of inductive biases used by machine learning algorithms.

- The inductive bias of a learning algorithm:
  (a) is a set of assumption about what the true function we are trying to model looks like.
  (b) defines the set of hypotheses that a learning algorithm considers when it is learning.
  (c) guides the learning algorithm to prefer one hypothesis (i.e. the hypothesis that best fits with the assumptions) over the others.
  (d) is a necessary prerequisite for learning to happen because inductive learning is an ill posed problem.
- An example of the specific inductive bias introduced by particular machine learning algorithms would be good here. E.g.:
  - 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. The assumption is that distinct classes tend to be separated by wide boundaries.
  - Minimum cross-validation error: when trying to choose among hypotheses, select the hypothesis with the lowest cross-validation error.

13. Explain what can go wrong when a machine learning classifier uses the wrong inductive bias.

- If the inductive bias of the learning algorithm constrains the search to only consider simple hypotheses we may have excluded the real function from the hypothesis space. In other words, the true function is **unrealizable** in the chosen hypothesis space, (i.e., we are **underfitting**).
- If the inductive bias of the learning algorithm allows the search to consider complex hypotheses, the model may hone in on irrelevant factors in the training set. In other words the model with **overfit** the training data.

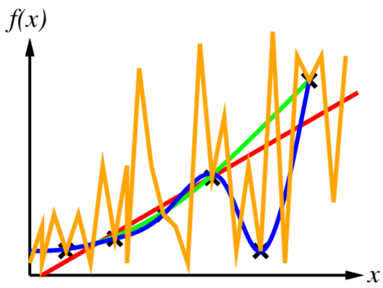
14. Why is it difficult to select the correct inductive bias for a machine learning algorithm?

- Choosing the correct inductive bias for a particular inductive learning problem is difficult because it involves getting the balance right in the **tradeoff** between the expressiveness of a hypothesis space and the complexity of finding a simple, consistent hypothesis within that space. In other words, there is a tradeoff between complex hypotheses that fit the training data well and simpler hypotheses that may **generalise** better (How well a model trained on a training set predicts the right output for new instances is called **generalization**).
- If the hypothesis class $\mathbf{H}$ is less complex than the true function we are trying to model, we have **underfitting**. In other words the inductive bias is to restrictive on the search and the learning algorithm is not able to generate hypotheses that are complex enough. In the case of underfitting, as we increase the complexity allowed by the inductive bias, (i.e., increase the complexity of $\mathbf{H}$) the training error decreases.
- If our model selection has chosen a $\mathbf{H}$ that is too complex, the data is not enough to constrain it and we may end up with a bad hypothesis. Whenever there is a large set of possible hypotheses, one has to be careful not to use the resulting freedom to find meaningless "regularity" in the data. This problem is called **overfitting**. Overfitting becomes more likely as the hypothesis space and the number of input attributes grows, and less likely as we increase the number of training examples.

15. Inductive machine learning is often referred to as an *ill-posed problem*. Explain why this is the case and discuss the implications that arise because of it. In your answer be sure to refer to examples of specific inductive machine learning algorithms.

This is a fairly open question and so a rigid marking scheme is inappropriate. However, to score highly students should cover the following points:

- What is meant by an ill-posed problem and what are the implications of this for machine learning.
  - Inductive machine learning algorithms essentially search through a hypothesis space to find a the best hypothesis that is consistent with the training data used. It is possible to find multiple hypotheses that are consistent with a given training set (i.e. agrees with all training examples). It is for this reason that inductive machine learning is referred to as an ill-posed problem as there is typically not enough information in the training data used to build a model to choose a single best hypothesis. Inductive machine learning algorithms must somehow choose one of the available hypotheses as the *best*. An example like that shown in the figure below would be useful at this point



- How do machine learning algorithms deal with the fact that machine learning is ill posed.
  - Because inductive learning is ill-posed, we have to make some extra assumptions to have a unique solution with the data we have. The set of assumptions we make to have learning possible is called the **inductive bias** of the learning algorithm - this is the main implication of inductive machine learning being ill-posed.
- Define what is meant by inductive bias:
  - The inductive bias of a learning algorithm:
    (a) is a set of assumption about what the true function we are trying to model looks like.
    (b) defines the set of hypotheses that a learning algorithm considers when it is learning.
    (c) guides the learning algorithm to prefer one hypothesis (i.e. the hypothesis that best fits with the assumptions) over the others.
    (d) is a necessary prerequisite for learning to happen because inductive learning is an ill posed problem.
  - An example of the specific inductive bias introduced by particular machine learning algorithms would be good here. E.g.:
    - 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. The assumption is that distinct classes tend to be separated by wide boundaries.
    - Minimum cross-validation error: when trying to choose among hypotheses, select the hypothesis with the lowest cross-validation error.

- The 2nd half of the solution this question continues in the next answer box.

- This is the 2nd part of the solution to: "Inductive machine learning is often referred to as an *ill-posed problem*. Explain why this is the case and discuss the implications that arise because of it. In your answer be sure to refer to examples of specific inductive machine learning algorithms.":

- The importance and difficulty of selecting the right inductive bias:
  - As learning is not possible without inductive bias the question becomes **how to choose the right bias?** This, however, is important and difficult because:
    - If the inductive bias of the learning algorithm constrains the search to only consider simple hypotheses we may have excluded the real function from the hypothesis space. In other words, the true function is **unrealizable** in the chosen hypothesis space, (i.e., we are **underfitting**).
    - If the inductive bias of the learning algorithm allows the search to consider complex hypotheses, the model may hone in on irrelevant factors in the training set. In other words the model with **overfit** the training data.
- The student should illustrate an awareness of: (1) the tradeoff in hypothesis consistency and generalizability, (2) and the related issues of issues of underfitting and overfitting.
  - There is always a **tradeoff** between the expressiveness of a hypothesis space and the complexity of finding a simple, consistent hypothesis within that space. In other words, there is a tradeoff between complex hypotheses that fit the training data well and simpler hypotheses that may **generalise** better (How well a model trained on a training set predicts the right output for new instances is called **generalization**).
  - If the hypothesis class $\mathbf{H}$ is less complex than the true function we are trying to model, we have **underfitting**. In other words the inductive bias is to restrictive on the search and the learning algorithm is not able to generate hypotheses that are complex enough. In the case of underfitting, as we increase the complexity allowed by the inductive bias, (i.e., increase the complexity of $\mathbf{H}$) the training error decreases.
  - If our model selection has chosen a $\mathbf{H}$ that is too complex, the data is not enough to constrain it and we may end up with a bad hypothesis. Whenever there is a large set of possible hypotheses, one has to be careful not to use the resulting freedom to find meaningless "regularity" in the data. This problem is called **overfitting**. Overfitting becomes more likely as the hypothesis space and the number of input attributes grows, and less likely as we increase the number of training examples.
- students should refer to examples of how specific inductive machine learning algorithms overfit or underfit - e.g. overly complicated decision trees, regression approaches that use polynomials of too high/low a degree, neural network algorithms that train for too long leading to an overly complicated decision boundary.
- The implications of inductive machine learning being an ill-posed problem can be summarised through *Dietterich's triple trade-off.* The triple trade-ff states that all inductive learning algorithms there is a trade-off between three factors:
  (a) the complexity of the hypothesis space $\mathbf{H}$ we fit to the data,
  (b) the amount of training data,
  (c) the generalisation error on new examples.

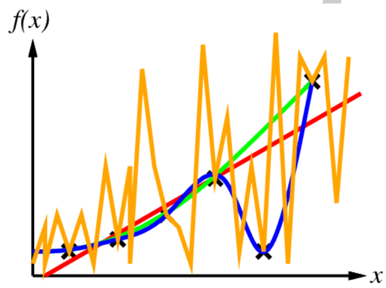
16. *Dietterich's triple trade-off* refers to the fact that in all inductive machine learning algorithms there is a trade-off between the following three factors:

- the complexity of the hypothesis space we fit to the data
- the amount of training data
- the generalisation error on new examples

Discuss the origins of this trade-off and its implications. In your answer be sure to refer to examples of specific inductive machine learning algorithms.

This is a fairly open question and so a rigid marking scheme is inappropriate. However, to score highly students should cover the following points:

- Inductive machine learning algorithms essentially search through a hypothesis space to find a the best hypothesis that is consistent with the training data used. It is possible to find multiple hypotheses that are consistent with a given training set (i.e. agrees with all training examples). It is for this reason that inductive machine learning is referred to as an ill-posed problem as there is typically not enough information in the training data used to build a model to choose a single best hypothesis. Inductive machine learning algorithms must somehow choose one of the available hypotheses as the *best*. An example like that shown in the figure below would be useful at this point



- Because inductive learning is ill-posed, we have to make some extra assumptions to have a unique solution with the data we have. The set of assumptions we make to have learning possible is called the **inductive bias** of the learning algorithm The inductive bias of a learning algorithm:

  (a) is a set of assumption about what the true function we are trying to model looks like.

  (b) defines the set of hypotheses that a learning algorithm considers when it is learning.

  (c) guides the learning algorithm to prefer one hypothesis (i.e. the hypothesis that best fits with the assumptions) over the others.

  (d) is a necessary prerequisite for learning to happen because inductive learning is an ill posed problem.

- An example of the specific inductive bias introduced by particular machine learning algorithms would be good here, e.g.:

  – 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. The assumption is that distinct classes tend to be separated by wide boundaries.

  – Minimum cross-validation error: when trying to choose among hypotheses, select the hypothesis with the lowest cross-validation error.

- The 2nd half of the solution this question continues in the next answer box.

- This is the 2nd part of the solution to the question about the origins of *Dietterich's triple trade-off* and its implications:

- As learning is not possible without inductive bias the question becomes **how to choose the right bias?** This, however, is important and difficult because:
  - If the inductive bias of the learning algorithm constrains the search to only consider simple hypotheses we may have excluded the real function from the hypothesis space. In other words, the true function is **unrealizable** in the chosen hypothesis space, (i.e., we are **underfitting**).
  - If the inductive bias of the learning algorithm allows the search to consider complex hypotheses, the model may hone in on irrelevant factors in the training set. In other words the model will **overfit** the training data.
- This is essentially Dietrich's triple trade-off. There is always a **tradeoff** between the expressiveness of a hypothesis space and the complexity of finding a simple, consistent hypothesis within that space. In other words, there is a tradeoff between complex hypotheses that fit the training data well and simpler hypotheses that may **generalise** better (How well a model trained on a training set predicts the right output for new instances is called **generalization**).
- Underfitting
  - If the hypothesis class $\mathbf{H}$ is less complex than the true function we are trying to model, we have **underfitting**
  - In other words the inductive bias is to restrictive on the search and the learning algorithm is not able to generate hypotheses that are complex enough.
  - In the case of underfitting, as we increase the complexity allowed by the inductive bias, (i.e., increase the complexity of $\mathbf{H}$) the training error decreases.
- Overfitting
  - If our model selection has chosen a $\mathbf{H}$ that is too complex, the data is not enough to constrain it and we may end up with a bad hypothesis.
  - Whenever there is a large set of possible hypotheses, one has to be careful not to use the resulting freedom to find meaningless "regularity" in the data.
  - This problem is called **overfitting**.
  - Overfitting becomes more likely as the hypothesis space and the number of input attributes grows, and less likely as we increase the number of training examples.
- Students should refer to examples of how specific inductive machine learning algorithms overfit or underfit - e.g. overly complicated decision trees, regression approaches that use polynomials of too high/low a degree, neural network algorithms that train for too long leading to an overly complicated decision boundary.