IOC Topic 5.1 - Introduction to Machine Learning

Transcript & Notes: PART 2

Author: Dr. Robert Lyon

Contact: robert.lyon@edgehill.ac.uk (www.scienceguyrob.com)

Institution: Edge Hill University

Version: 1.0

Topic 5, Module 1, Part 2

Introduction Slide

Hello and welcome back to Topic 5, Module 1, Introduction to machine learning. This is part 2 of the module. It will gently introduce ideas some basic mathematical ideas underpinning machine learning, and your first "intelligent" algorithm. Part 2 will take approximately 1 hour to complete. My name is Dr. Robert Lyon, and I'll be your guide during this module. Notes will be provided that accompany this content. I advise you to keep those near you as we move through the slides. Each slide is numbered, and this number corresponds to a page in the notes.

In the notes I'll sometimes encourage you to undertake some optional self-study. These self-study opportunities will help you understand the content presented. I'll also provide links and references in the notes that enrich the content being discussed – follow those up at your leisure.

Slide 2

Let's take a moment to review the topics this module will introduce…

- Useful terminology and key concepts. This will help you understand more advanced content as we progress.

  This content has been covered in part 1.

- The mathematical background required to understand machine learning. This will be basic – the focus is on fostering an understanding, without worrying about complex equations. Our first automated learning system will follow shortly after.

  This content will be covered here in part 2.

- A number of machine learning algorithms from first principles, supported by examples you can try for yourself.

  This content will be introduced in part 3.

The aim is to help you acquire the foundational knowledge required to apply machine learning in practice.

Let's continue our journey by considering the information use for decision making more closely.

Slide 3

Many have tried to understand how to make optimal decisions. We know we should use available evidence at all times.

Humans are not always so thorough - we do make bad decisions. We are biased decision makers - we often use instinct and personal experience to decide.

We can visualise our knowledge and experience which helps us understand decision making more clearly. Suppose the blue box labelled 1, represents the set of all knowledge of some subject. This idealised set that contains every piece on information available which may be stored in books, web resources or in the minds of other people.

The circle labelled 2, represents the portion of that knowledge that is available to us. It is an incomplete portion; thus, it is a subset of the knowledge available. It is incomplete for many reasons. For instance, our local library doesn't contain every book ever written on a subject. Nor do we have access to the knowledge of scientists working at the forefront of research. So, we already have a gap in our knowledge at this step.

Then there is the personal knowledge and experience of the subject that we already possess. This is represented by the circle labelled 3. This is a subset of the knowledge available to us. This will vary from person to person. For example, if the subject is planet formation, those with PhDs in Astronomy will have greater knowledge than those without such a background.

Finally, there is the knowledge we ultimately use to make a decision. This is represented by the circle labelled 4. This is a subset of the knowledge we possess. For example, when trying to classify unknown animals as either birds or mammals, we don't need to use our knowledge of insects.

Here we've talked about knowledge, but not the mechanism via which it is used to make decisions. In humans that cognitive function is sophisticated. Emotions and personal bias influence our interpretation of available information, which in turn impacts our decision making.

Let's consider 1 more example. Over thousands of years the human race has conducted countless scientific experiments showing the Earth is round. We've collated mounds of information related to our planet, it's properties, and its place in the solar system.

Yet there are groups of people who reject this information due to their personal biases and how they interpret the world around them. They erroneously believe the world is flat. More knowledge will not necessarily change their view – rather their biases and thinking process must change.

Additional Notes:

This Wikipedia article provides a list of cognitive biases – how may do you suffer from? https://en.wikipedia.org/wiki/List_of_cognitive_biases

It may interest you to know that there are "Flat Earth" societies:

https://en.wikipedia.org/wiki/Modern_flat_Earth_societies

Slide 4

How does this view of human knowledge and decision making relate to automated machine learning? Well, machine learning algorithms actually make decisions similarly to humans. This means they are subject to the same problems of bias, and flawed decision making. However, the means by which this happens is quite different.

This is because Machine Learning is concerned with making optimal decisions primarily using the tools of statistics.

Artificial systems receive features taken from sensors (e.g. cameras, microphones etc.) or data from some other computational/record system that keeps track of information. Much like in humans, where experiences are associated with outcomes, the input features are associated with some outcome or label. This is represented by the set E.

To make decisions and predictions using this information, machine learning uses statistics. This type of mathematical decision making uses only the evidence available. You may think that this makes automated learning immune to bias. This is unfortunately not the case. Machine learning can be just as fallible for two main reasons.

Slide 5

The first reason automated decision making can be flawed - biased input data.

- For example, suppose you want to teach an algorithm to recognize a specific disease in a group of patients.
- You provide the algorithm with data, which just so happens to describe patients aged between 60 to 80.
- You then run the algorithm on patients aged 18 to 30. The algorithm performs badly, as its knowledge is biased toward recognizing disease in much older patients.

The second reason is biases in the cognitive process. For instance, suppose we try to teach an algorithm to predict when a train on the Tokyo rail network will be late.

- Trains on this network are exceptionally punctual.
- To achieve the best overall performance, just never predict that a train will be late.

Slide 6

We've reviewed some of the issues faced when trying to learn using machine learning. Let's recap:

- Algorithms can process more quantifiable data than humans. They can process entire databases or libraries of information, if it is presented in the right way.
- However, the data may be biased/incomplete. We must ensure the data we give to our algorithms is not biased, or incomplete such that it can give a misleading impression.
- Individual algorithms also have intrinsic biases. We must keep that in mind when considering what we are trying to do with machine learning.
- Algorithms can therefore be just as fallible as humans! There are many examples of this. In truth, our algorithms are only as good as we make them. Thus, in reality poor automated decision making reflects our own deficiencies more than anything.

Let's bring this together. Here the blue box labelled 1, represents the set of all knowledge of some subject available. This is an idealised set that contains every piece on information available which may be stored in books, web resources or in the minds of other people.

The circle labelled 2, represents the portion of that knowledge that is available to an algorithm. It is an incomplete portion; thus, it is a subset of the knowledge available. It is incomplete for many reasons. For instance, available data hasn't been digitised, or features are yet to be designed and extracted. We already have a gap in our knowledge at this step.

This circle labelled 3 represents the data that has been digitised and turned into features.

Then there is the knowledge and experience captured by the features, represented by circle 4. Features are not perfect; knowledge is lost here.

Finally, there is the knowledge and experience captured by the algorithm from the feature data. This is represented by the circle labelled 5.

Slide 7

- In machine learning the experience used by an algorithm to learn is known as "training data".
- An algorithm is 'taught' using training data. It uses the examples and labels in the training data, to solve problems using the tools of maths and statistics.
- Training data can be labelled (see the notes for Part 1, Slide 7) or unlabelled (see the notes for Part 1, Slide 9), which is something we learned in Part 1 of the module.
- To check that our algorithms have successfully learned what we wished them to from the training data, we must test them. To do this we evaluate them on a new set of data. We call this "test data".
- Test data must be labelled (that is, the ground truth must be known). Test data must also be distinct from training data. This is crucially important. Doing this helps ensure that our algorithms don't overfit to the training data – much like the student in Part 1 of the module that failed their exam (see the notes for Part 1, Slide 16).

Slide 8

There is an easy way to understand the difference between training and test data.

- Here the blue box represents the set of all knowledge of some subject available. This idealised set that contains every piece on information available which may be stored in books, web resources or in the minds of other people.
- Training data represents the knowledge given before an exam.
- Test data is a disjoint set of information, that represents the exam – we use this to test performance.

Slide 9

Whilst humans can automatically extract sophisticated features to make decisions, our artificial analogues are much less capable.

Generally, we must extract features for them. Whilst a human can look at an animal and automatically assess its mass (approximately), we must give the artificial system the mass value directly. But how is this information used?

In humans, you can scan your experience and consider examples of animals with similar masses, to an unknown animal currently in front of you.

Artificial systems instead compare the mass value against a numerical distribution (stored in $\boldsymbol{E}$) describing the mass values for all the animals previously observed. It can then connect the distributional information to the labelled feedback at its disposal - i.e. choose a label appropriate for an animal of this mass.

There are some methods that can extract features for themselves. However, this is an advanced topic that may be covered in future modules.

We must also provide useful labels in the training and test sets – again these are designed by us to achieve some specific purpose.

Feature design and extraction is an important and necessary step that must be taken, before we create our training sets. You may wonder how to design features, or if there's some accepted process for doing so?

Slide 10

- In principle feature design involves studying data.
- This requires thinking about its properties. What is the "information content" of the features? For example, when classifying mammals and birds, using a feature such as "number of eyes" may not be useful – all have 2. Such a feature has low information content. We desire features with high information content.
- Extracting information, you believe will help with class separation. Features with high information content help with class separation. A feature such as mass will have higher information content when trying to separate birds and mammals then "number of eyes". Though the feature "has feathers", is of much high information content – it can be used to perfectly separate the two groups.
- In practice feature design involves,

1. Considering as many candidate features as possible. Collect all the features you could use.
2. Considering their usefulness in turn.
3. Selecting a sub-set of the best features that you think will be most effective.
4. Testing those selected, and checking what happens when we use them to train and test a classifier.
5. The features chosen may turn out to be poor. So, we may need to return to step 1.
6. Collect new data? Maybe the data you have is simply not good enough.
7. Deriving new features? Perhaps the features you have a poor, but can they be combined in some way that makes them useful? Can you think of a feature that would be useful for separating Felines and Canines? What about a feature that equals mass divided by height – would that be useful? Not all Felines are small, and not all Canines are large. So, this is unlikely to be as effective as we'd like. What about a body length divided by tail length? Do Felines have longer tails relative to their bodies then Canines? Have a think about these questions in your own time.

Additional Notes:

Above I suggest the feature: body length divided by tail length.

This actually represents the following hypothesis:

I hypothesise that there is a difference in tail length relative to body length, that can help distinguish Felines from Canines.

Is this hypothesis correct? I don't know for certain - it is untested. Many features stored in real-world datasets represent hypotheses that may not have been tested. It is up to us as practitioners of machine learning, to ensure our features are robust and valid. If we don't stay vigilant, things can go very wrong. Systems can become inaccurate and flawed. Unfortunately, this has already happened, with two specific groups bearing the consequences due to Gender and Racial bias. Read more about some of these examples here:

https://medium.com/thoughts-and-reflections/racial-bias-and-gender-bias-examples-in-ai-systems-7211e4c166a1

Slide 11

We've covered a lot of material so far. We're now ready to review the whole classification process.

1. First, we're given a data set relating to some problem we wish to solve.
2. We design and extract features from this dataset.
3. This allows us to form a training set that we can use to teach a machine learning algorithm. We can also form a test set that we use to evaluate what the algorithm has learned. When satisfied that the algorithm has learned effectively, we can move on.
4. We're then given a new data set of unseen examples, that are not accompanied by labels. The ground truth is not known.
5. We extract features from this dataset. These are the same features we extracted when forming the training set.
6. We collect this new input data, and pass it to our machine learning algorithm. We ask it to make some predictions on this input dataset.
7. The algorithm outputs the input data but with true class labels – these are the predictions.

Slide 12

So far, we've learned all about the process of learning, but not how it's actually done.

In machine learning this is achieved via trial and error on the training set – learning from the mistakes made. This is best understood via an example.

We need to be at work, but don't know what time the bus arrives to get us there. Work starts at 09:00, and the journey takes roughly 40 minutes. We start to wait at the bus stop each day, recording some details.

Here we see some data recorded over three days. It shows what time we arrived at the bus stop, what time the bus arrived at the bus stop, and whether or not we were late for work. This represents training data. Each row represents a trial, and the label lets us know if we made it to the bus stop early enough.

In this case learning involves finding an arrival time at the bus stop, that gets us to work on time. Each time we're late, we've made an error that we learn from.

Slide 13

In ML error is quantified and minimised using the tools of mathematics.

We can reduce finding the best time to wait at the bus stop, to a mathematical problem, i.e. find a value for $\boldsymbol{t}$ that minimizes the number of times we're late.

Many potential values for $\boldsymbol{t}$ work, but clearly it must be 08:15 or earlier!

This sort of error minimisation is done using what we call "functions". Before proceeding, I advise that you watch the tutorial video (11 minutes long) which I provide the link to in the notes:

https://www.youtube.com/watch?v=52tpYl2tTqk

Slide 14

Functions can be thought of as simple input/output boxes.

Machine learning algorithms are functions, or combinations of many functions, that ingest data, and produce some output.

How they do this is hard to convey without some basic mathematics!

For instance, here is a function that accepts two inputs, x and y.

$$(x, y) = x + y$$

It returns an output value that is simply equal to x plus y.

If we have $x = 2$ and $y = 4$ this function, simply returns an output value of 6.

$$f(x = 2, y = 4) = 2 + 4 = 6$$

Additional Notes:

A gentle introduction to functions: https://www.mathsisfun.com/sets/function.html

Slide 15

Suppose we have some input data that contains features describing patients that recently had a blood test.

We desire a function that can accurately assign labels to patients indicating whether or not they're diabetic.

What we need is a mapping function. This is a function that maps input values in X, to the correct labels in Y.

A simple function would be something that splits the data on a single feature. For example, if the value of feature 1 is less than or equal to 0, then predict 0, which is the non-diabetic label, otherwise predict 1. Here learning involves finding a feature, and a split value over that feature, that achieves the fewest errors possible.

Slide 16

Here's a concrete example showing patient data. These patients have been subject to a test for diabetes. This test is called an oral glucose tolerance test.

During this test, the patients have a morning blood test prior to eating. Ideally the patient will not have had any food or drink for 8 to 10 hours.

The patient is then given a glucose drink. Two hours after ingesting the glucose drink, another blood sample is taken. Medical professionals then interpret the results and treat the patient s accordingly.

There are 7 features in total in the table. These include gender, age, weight (Kg), height (cm), a measurement of their glucose level in the morning, and their 2-hour fasting glucose measurement.

There are two class labels that can be assigned –non-diabetic represented by 0, and diabetic represented by 1.

It's hard to look at this data and determine what characteristics make someone diabetic. There's a lot of information to look through. Suppose we decide to test how useful 2 arbitrarily chosen features are for this classification task.

Age and height are chosen, and we plot those feature values in a 2-d plot which is now shown. We can see the diabetic patients represented by red dots, and the non-diabetic by the blue. Can we use a simple function to split these patients accurately into diabetic and non-diabetic classes?

We could use a function that splits on the Height feature – that involves choosing a split value over height and assigning labels based on that. We can do the same for age.

Slide 17

Let's try Age first. We know that age is unlikely to be a good feature, as people of all ages are susceptible to diabetes. However, we'll try age, as we known that as people age, they can be more susceptible to developing the illness.

What value for age gives us a split that produces the best classification accuracy in this example?

We could choose any age, but the optimal choice here, is to split when $age = 34$. We assign patients on the left-hand side of the line the diabetic label, and those on the right-hand side the non-diabetic label. Using this separating line defined by $age = 34$ results in 5 mistakes. One non-diabetic person is incorrectly classified as diabetic, and four diabetic people are incorrectly labelled as non-diabetic.

This split point produces a line that separates the data. We call this line a decision boundary.

Slide 18

We can visualise the method used to split this data using the age feature.

Either assign the diabetic label if age is less than or equal to 34…

Otherwise predict non-diabetic.

This example may seem entirely contrived. However, we've actually just encountered our very first machine learning algorithm. It's called a "Decision Stump".

A Decision Stump is a linear separator, which can also be described as a linear model.

It simply looks for the best feature to split upon, based on the information in the training set. It uses the labels in the training set to guide the feature search, and to find the best split value.

The Decision Stump produces a single linear separator. This means it is only useful for problems where there are two classes which may be separable using a linear decision boundary.

Slide 19

The learning process for the decision stump is simple.

For each feature in the data, the stump will search for a threshold across all features, that minimises the error rate.

We can see how the stump makes fewer and fewer errors, as it moves toward the optimal split value.

In our example we used age as a feature, and our split-value = 34. There are better ways to split this data accurately. However, the Decision Stump is too simple to find those solutions. Only more complex methods that produce non-linear decision boundaries, can solve this problem.

Slide 20

You've now seen a linear model in action and learned what a decision boundary is.

Real-world problems are complex - so too are the decision boundaries needed to accurately separate data.

This means linear models don't always work well. More complex ML algorithms are required. The image to the left visualises the same patient data before using different features. We can see a decision boundary represented by the black line, that effectively classifies the data. The data points that fall within the shaded region are classified as diabetic, all other data points are labelled non-diabetic. This boundary achieves optimal accuracy on the training data.

When we generalise well we can produce effective decision boundaries such as this. However sometimes we don't perform well in practice.

We can underfit to our data (see the notes for Part 1, Slide 17). Here we can see a new decision boundary represented by the dashed blue line. Now examples falling into the blue shaded area are classified as diabetic, and all other data points as non-diabetic.

We overfit to our data (see the notes for Part 1, Slide 17). Here we can see a new decision boundary represented by the dashed red line. Now examples falling into the red shaded area are classified as diabetic, and all other data points as non-diabetic. This boundary achieves perfect accuracy, but this will not transfer well to new data.

Decision boundaries allow us to visualise this happening, so we can guard against it. Ideally, we need boundaries that generalise well to new data.

Slide 21

We've covered these topics in the last few slides:

- Learning from a training set.
- Using a test set.
- That humans and algorithms suffer from bias, inherent to the data available to them.
- The classification process.
- Functions, and mapping functions in particular.
- The concept of error minimization.
- Decision boundaries.
- Our first ML algorithm, the decision stump.

In the next part we'll encounter some new algorithms and continue building up our knowledge.