School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (Semester 1, 2021)
Sample Solution Week 2

Considering the following problems:

(i) Building a system that guesses what the weather (temperature, precipitation, etc.) will be like tomorrow

(ii) Predicting products that a customer would be interested in buying, based on other purchases that customer has previously made

(iii) Skin cancer screening test

(iv) Automatically identifying the author of a given piece of literature

(v) Finding the best burrito in the United States of America

1. Identify the "concept" we might attempt to "learn" for each problem (Task Identification)

What we trying to learn in usually the parameter (or concept) that we are trying to predict or understand (using a Machin Learning technique). It is the final output of the system which can be a label (such as sunny, rainy, cloudy) or a quantity (like the possible temperature) or a cluster (like spam / not-spam) or something else (e.g., an association rule). In a *supervised learning* problem (such as classification or regression) this concept usually referred to as a *label* or the *response variable.*

As for our sample problems:

(i). Various weather features of the particular day (that we are trying to predict) can be considered as the output of the system. The prediction can be a quantity like the temperature or amount of rain or the UV index or any other weather feature.

(ii). There are two approaches to this problem:

a. We want to exhaustively label every product for every customer as either "interested" or "not interested". For example, we know that for the past 6 months the customer purchased 'milk' (in average) every 7 days. So, we can predict if the customer would be "interested" in purchasing 'milk' next time (s)he enters the store.

b. We want to predict if our customer would be interested in a single product (or set of products) that (s)he has not purchased before. In this situation we can find the group of customers that have similar purchase habits/taste and based on their purchase history predict the behaviour of our particular customer.

(iii). Since it is a screening test, it is clear that we are trying to answer whether a patient has a cancer or not. It is a binary decision (True or False) which it is a very common in Machine Learning.

(iv). The simple answer is that we are trying to find the "author" of a writing, so that's the concept we are trying to learn (predict). However, depending on the domain of the problem, the approaches can be different.

(v). The example in (d) might seem whimsical, but this was actually attempting somewhat seriously (https://fivethirtyeight.com/tag/burrito-bracket/ — you might like to examine their study design and features). The key question here is that we are not actually looking for a single unique burrito

that we can hold in our hands and say that it is truly the “best” one (whatever that means), but rather a particular restaurant (or product from a restaurant) that is consistently “better” than comparable products from other restaurants.

2. For each problem-task, identify what the instances and attributes might consist of (choosing the data representative)

   An instance is a single exemplar from the data, consisting of a bundle of (possibly unknown) attribute values (feature values) [and in the case of supervised ML a class value].

   An attribute is a single measurement of some aspect of an instance, for example, the frequency of some event related to this instance, or the label of some meaningful category.

   Attributes are usually classified as either nominal (labels with no ordering), ordinal (labels with an ordering), or continuous (numbers, even if they perhaps aren’t continuous in the mathematical sense).

   (i). It seems fairly clear that each instance will be a day; depending on how we construe the problem, various properties could be attributes — the most logical is probably the corresponding data (temperature, precipitation, humidity, wind speed, etc.) from the previous day(s).

   (ii). For scenario (a) of last question, customer-product pairing can be the possible instance. For scenario (b) each customer would be an instance. In either way the attributes can be the customer’s name, age, address, gender, shopping log, credit card information, loyalty card information and more.

   (iii). In this case each patient is an instance. The attributes can be results of the blood test, images from the skin, reports, observed syndromes and so on.

   (iv). Here we can have different situations:

      a. We can have a single unknown piece of literature and a fixed set of authors who may have written it (and a collection of their previous writing). Here the instances are the writings, and the attributes can be the words or grammatical structures, and perhaps some metadata (such as year of publication, language, publisher, and so on)

      b. If we have an open–domain problem — that potentially anybody could have written it, then the instances are still the same. But we may need to use other attributes (such as linguistic properties) as well.

      c. We might instead have a situation like plagiarism detection, where we don’t have access to many data for any individual author. In this case the attributes again can be the words or grammatical structures. In this case, we might want to treat individual paragraphs, or even individual sentences as the "instances" rather than the whole piece of literature. That way we wouldn't have to do "one shot" learning on just one document per author; instead, we'd learn from many paragraphs per author. And it seems like a more natural way to represent the problem, since each paragraph (or each sentence!) in the suspect document could have been stolen from a different source author.

   (v). Here we can also have two approaches:

      a. Considering the product (burrito of a restaurant) as the instance and use features like ingredients, sauces, spices and so on.

      b. We can consider the restaurant (that sells Burrito) as the instance and use features like the ranking of the restaurant or the customers compliments (that mentioned Burrito) as features.

3. For each problem-task, conjecture whether a typical strategy is likely to use supervised or unsupervised Machine Learning (picking a suitable model)

   Generally speaking, supervised techniques in machine learning start from exemplars (instances) — labelled with classes — in a set of training data and use these to classify unknown instances in a set of test data.

   Unsupervised methods are not based on a set of labelled training data. Unsupervised methods often broken down into 'weakly unsupervised methods' (where the class set is known, but the system does not have access to labelled training data), and 'strongly unsupervised methods' (where even the class set is unknown, and we don't even know how many classes we have).

   (i). For this problem, assuming that we can access historical data for the particular location, (supervised) **regression** seems like the most plausible ML strategy. So we find the pattern using the attributes value from previous days, months and years and predict our weather feature (e.g. temperature). This case could potentially also be **classification** – instead of predicting the temperature, wind speed, etc. we can just give one label like on a weather app ("Sunny," "Rainy," etc.).

   (ii). For our two different approaches:

      a. In this scenario, we have a **classification** problem, where we might try to predict "interested" "not-interested" labels based on some properties of the product and customer. Classification is a supervised learning method.

      b. It can be a (unsupervised) **clustering** method, where we find groups (clusters) of customer with same features; or an **association rule mining** method that we identify an association between customer(s) and some attribute(s) in the products. (e.g., if the product is from 'Nestle' there is x% probability that customers age groups of A and B would purchase it.)

   (iii). Assuming that we have trained our model based on the historical data from previous patients, it would be a (binary) **Classification** problem.

   (iv). For our three different scenarios:

      a. If we have a single unknown piece of literature and a fixed set of authors who may have written it — and a collection of their previous writing — then this is probably a **classification** problem, where we might associate each piece of writing with the words (or grammatical structure, and perhaps metadata) contained within it.

      b. If we have an open–domain problem — that potentially anybody could have written it — then collecting labelled data would be possible (i.e., classification), albeit obnoxious. We might instead prefer to use a **clustering approach** based on the document's linguistic properties (although this is unlikely to identify a single author).

      c. In case of plagiarism, simple classification is unlikely to be very effective (because our model might be insufficient to represent each author), but we could try something like **outlier detection** or **semi-supervised learning** (which we'll talk about later in semester) to detect "probably plagiarised" sections in any document. If we treat sentences or paragraphs as instances, **classification** might as well be possible -- we have limited data, but we can look for a near-exact match in our resources (pieces of writings from original authors that sentence (or paragraph) may been copied from).

   (v). It's a **classification** problem.

4. For each problem-task, consider how easy or difficult it would be to make a model that generalizes to new cases. For example, could you predict the weather in any city in the world, or just in one specific city?

(i). The weather model might work better in some cities than others. It would probably generalize better if it included geographic information in addition to the previous days' weather (e.g., longitude, altitude, distance from ocean, distance to mountains) because then it could learn how these features interact with the weather patterns.

(ii). A customer model trained in one country might not generalise to other countries. If it mostly learns everyday shopping patterns, it probably won't give good predictions for outlier situations like holiday purchasing.

(iii). Generalization is a big concern for machine learning in the medical domain, because real world training data often have biases, and these biases can affect performance in various ways. For example, we know skin cancer risk increases with age, so these variables will probably be correlated in your training set. On the one hand, this is good – the model should correctly learn that age predicts skin cancer. But it can also be bad if the model becomes too dependent on that predictor (e.g., if it decides to label every image of older-looking skin as "cancer"). And if there were very few instances of young people with cancer in the training set, the resulting model might not work well on younger patients.

(iv). In the classification example, you would hope the model would generalize well (e.g., after seeing many examples of "Shakespeare" and "Marlowe" it could reliably classify new examples from each author), but you would probably need to train a new model for a new set of authors (e.g., "Shakespeare" or "Burbage"). In the case of plagiarism / outlier detection, you might be able to generalize to new cases – you might learn some general rules to decide "what is an outlier?" even though the exact outlier features might be different for each author.

(v). How well the model generalizes would depend on what attributes you use and what attributes you care about in a "best X" problem. A "best burrito" model might also be able to pick the "best pizza" because the predictive attributes are similar (good taste, good value, quick service, etc.). But it might not predict "best coffee" since people care about other factors like the café atmosphere, noise levels, and study space which aren't relevant to the burrito problem.

5. What kinds of assumptions might a machine learning model make when tackling these problems?

Every model makes assumptions about the world and how the concepts we want to learn relate to the attributes of the data.

The first assumption we make is that the concept is actually related to the attributes! This assumption is so obvious that we rarely discuss it – usually we only include attributes that we think are likely to predict the concept. For example, you would probably not use "patient's favourite song" as an attribute for skin cancer detection. However, this attribute might actually be a good predictor, because your favourite song can be a good predictor of your age, and age is a risk factor for skin cancer. You could probably come up with other "weird" predictors for each of the example models.

Secondly, each model makes assumptions about the ways the attributes can relate to the concepts. For example, does it make more sense for the models to treat all attributes as independent predictors, or would it be better to use a model that allows the predictors to interact? In most of these cases we would expect the attributes to interact in complex ways but allowing interactions could lead to an overly complex model in the cases where there are many attributes to start with (for example, in the customer purchasing model). For the problems with numeric attributes, would we generally expect linear (or monotonic, e.g., strictly increasing or decreasing) relationships between the attributes and concepts. This is often a good simplifying assumption for machine learning, but it limits what a model can learn. For example, the relationship between "best burrito" and price might be U-shaped – very cheap and very expensive burritos might be less popular than burritos priced somewhere in the middle.