Musical Instrument Accessibility

Diana Whealan, ECE '19

Introduction

Machine Learning has emerged in recent years as a powerful analytical tool. Its value comes from its ability to continually learn from the information given to a program. A traditional program follows specific steps, like instructions in a recipe. For example, lets say a baker is making some banana bread. He or she will cook the bread for the time specified in the recipe, however, if at the end of the allotted time, the bread doesn't look fully cooked, the baker will look at the banana bread and and know that it needs to cook for another 5 minutes. The baker might also know that the oven he or she is using is a little bad at regulating temperature, and will add the 5 minutes to every baking time. A traditional program would follow the instructions, or recipe, exactly, even if that means undercooked bread. A Machine Learning program can change based on information it is given, and can even recognize patterns, like an oven that takes longer to bake.

There are many considerations when designing a Machine Learning program. This paper will discuss these considerations, focusing on the objectives of Machine Learning, learning strategies, and the design of learning programs.

Objectives of Machine Learning

There are three main objectives of Machine Learning. The first objective is Task-Oriented Studies, and is also known as the *engineering approach.* The purpose here is to use Machine Learning to improve the performance of a predetermined set of tasks (Michalski, Carbonell, & Mitchel, 1983).

Engineers are problem solves. They tend to focus on the practical application of things. However, the second objective of Machine Learning does not focus as much on the practical. The Theoretical Analysis objective includes the desire to find all possible solutions that include Machine Learning, instead of just solutions that most closely replicate how humans would solve a problem.

The last object is Cognitive. A Machine Learning algorithm, the set of calculations executed by the program, often wants to both learn and make decisions as a human would, and thus becomes a core objective of the field of Machine Learning (Michalski et al., 1983).

Learning Strategies

We experience different ways of learning in school. Some teachers make their students memorize new vocabulary words, other require them to perform a skit using the new words. While Machine Learning algorithms don't perform skits, there are several ways an algorithm can learn form the information it is given. Machine Learning algorithm can be classified based on these learning strategies that range from the amount of decision making the algorithm must make outside of what is explicitly coded. Recall, a traditional program follows its instructions exactly and makes no *decisions* on its own. If a program is able to come up with new ideas, concepts, solutions, etc. then it can be assumed that the system is making inferences and learning form experiments and observations (Michalski et al., 1983). The various learning strategies have trade-offs between how much one must teach the system, and how difficult it is for the system to learn (Roy & Stewart, 2010).

Learning from Instruction

Learning from instruction is a form of learning that most directly parallels how students learn in a classroom. The students have a small amount of pervious knowledge, but rely on information from a textbook or other source of organized/structured knowledge, to add and enhance what he or she already knows. A Machine Learning algorithm of this nature would be able to accept instructions and apply them. While the system is making some decisions, it mostly relies on the instruction provided to effectively complete the task.

Learning from Examples

Learning from examples is a learning strategy that uses examples as well as counterexamples as a way of teaching a Machine Learning algorithm a general concept. After being provided with examples, the algorithm should have the ability to detect if something is a proper example, or if it is not. For example, by showing a child a book about cats, the child will start to learn what the cat is. If shown another animal, they will be able to recognize that it is not a cat. This technique requires more inference, or decision making, than that of learning from instruction. Learning from examples does to explicitly introduce the concept it wants the algorithm to learn. A programmer, who knows the concept he or she would like the algorithm to learn, can present the algorithm with positive examples, i.e. pictures of cats, negative examples, i.e. pictures that are not cats, or some mixture of both. The algorithm itself may also generate its own examples to further test and narrow its knowledge.

Learning from examples can be further categorized by when the system receives the examples. Examples can be given all at once, allowing a clearer path for what concept is to be learned. Examples can also be given incrementally, which is more similar to how humans learn. In this case, examples can be used to refine the desired concept (Michalski et al., 1983).

Learning by Observation

The last learning strategy that will be discussed is learning by observation and discovery. This strategy requires the most inference, as the algorithm is not told what concept it is learning or any instances of that concept, positive or negative. This allows the algorithm to be more flexible and could even lead to the algorithm learning several concepts. However, this can introduce problems when the algorithm cannot discern what observation correlates to which concept. For example, chemistry students often perform experiments to learn and solidify a new concept. Say they mixed two liquids together and they changed color and turned into a foam like substance. If the students were not previously told, they would be unsure if the concept they are learning is the changing color or the change from liquid to foam. This is the difficulty a Machine Learning algorithm trained through observation might have. It is called the “severe focus-of-attention problem” (Michalski et al., 1983).

Categorizing the different learning strategies allows engineers to compare them and chose the best one given the data available and desired objective

Designing of Programs that Learn from Examples

In order to design a program that can learn based on the examples it is given, the program must be able to define a definition, or group of statements, that capture every possible example. This is called a Maximally Specific Conjunctive Generalizations (MSC-generalization) (Watanabe & Rendell, n.d.). This definition is a generalization, meaning that several different examples would all be encompassed. This definition is maximal as it should encompass all examples that are desired. This definition is specific, as it will not allow non examples. Finally, this definition is Conjunctive, meaning it is the conjunction or connection of several statements. The definition is therefore the most detailed description of a group of objects.

Algorithms that learn from examples can be categorized into bottom-up or top-down. The bottom-up approach is where data is the driving factor. This approach takes all of the input into account as it generalizes a description of the data. For each new piece of data that is added, the description is broadened or made more specific until a final description, or MSC-generalization, is reached (Watanabe & Rendell, n.d.). The top-down method begins with a group of elements, or pieces of data, and creates a best guess, also known as a hypothesis, for what it thinks best describes them. The remaining elements, those that were not used to create the original hypothesis, are checked against the hypothesis to see if they fit into its description. The hypothesis is then adjusted if the element does not fit, or if the description can be made more specific. This method is more robust to anomalies in the data, but would take longer to run since the hypothesis must be repeatedly checked.

Conclusion

In Machine Learning there is no one way to do something. Trade-offs are inevitable. The programmer must decide what learning strategy as well as what algorithmic approach will lead to the most accurate predictions. These decisions are heavily based on the kind of data, or input, the program will receive.

For system that detects the deviation in piano key presses due to a disability, learning by example may be the best learning strategy. In this system, a positive example of pressing a key can be given. From here, examples that deviate from a "correct key press" can be detected. Additionally, one does not know all the ways in which someone will press the key incorrectly. For this reason, a bottom-up approach, that is data driven, would allow the concept of disability based incorrect presses can be developed from the examples given by the user.

The vast number of approaches to solving a problem using Machine Learning makes it the perfect engineering problem. There is no perfect solution, but through the evaluation of many techniques, one path forward can be chosen.

References

1. Michalski, R.S., Carbonell, J.G., & Mitchell, T.M. (1983). *Machine Learning: An Artificial Intelligence Approach.* Morgan Kaufman
2. Bishop, C.M. (n.d.). *Pattern Recognition and Machine Learning.* 9.
3. Watanbe, L., & Rendell, L. (n.d.). *1990-Effective Generalization of Relational Descriptions*. 7.
4. Wu, J., Roy, J., & Stewart, W. (2010). *Prediction Modeling Using EHR Data: Challenges, Strategies, and a Comparison of Machine Learning Approaches.*