Theory, Practice & Products

MInDS @ Mines

Three main approaches exist for work that happens in machine learning. The theoretical approach focuses on creating better models and proving how much better they are. The practitioner's approaches focuses on training a variety of models to a particular dataset and finding its best model. The product-based approach focuses on creating a usable product that provides a solution to a customer's real-world problem. We will cover the three approaches and what you need to know to work with them.

Machine learning is a large field with people from a diverse set of backgrounds working on it. Three main approaches exist to working with machine learning.

First, there's the theoretical approach which focuses on creating new models and mathematically proving their effectiveness. This approach is the one that creates new solutions to machine learning problems. Second, there's the practitioners' approach which focuses on solving a specific problem defined by a particular dataset. Practitioners work on understanding the intricacies of a dataset and finding the ideal model that can effectively represent the data. Third, the product-based approach focuses on creating a product that can solve a customer's problem using machine learning. The methods used to solve a user's problem are not important. Instead, the value is in the effectiveness of the solution produced and machine learning can be an effective part of that solution.

Figure 1: Spectrum of approaches. Note that Practice is closer to Product than theory.

These approaches resemble the steps that go into working with machine learning. First, we need to create models. We then determine the best one for our domain. Finally, we build a product that can effectively solve a customer problem. This perspective mimics the roles involved at a large organization providing a product that utilizes machine learning. Roles exist on a spectrum of the approaches and differ between organizations. Generally, we have:

The names, and seeming rigidity, of these approaches are not important. The core idea is understanding the different perspectives of the approaches. This is a spectrum where we can sometimes find some practitioners building products or working on creating new models as well.

1. Software Engineers - build a customer-facing product
2. Data Analysts - use data for decision making and to inform the rest of the organization or its clients
3. Data Engineers - create data pipelines and apply (optimized) algorithms to deliver machine-learning capabilities to the product
4. Machine Learning Engineers - Program machine learning algorithms interfacing with the product
5. Data Scientists - Determine the best machine learning algorithms that apply to the customer domain and fine tune them
6. Research Engineers - develop new machine learning algorithms and models relevant to the product domain [1]

[1] Research Engineers are primarily different from traditional researchers in the underlying goal or motivation of the research. Researchers are primarily motivated by the advancement of science while Research Engineers are (often) primarily motivated by the organization's bottom line.

Theory

In the previous lecture, we discussed the types of problems that machine learning can solve but not how they do so. The theory approach focuses on how machine learning models can solve the problems they aim to solve. Earlier, we defined machine learning as methods that learn rules from data and answers. Another way to define machine learning is *model parameter estimation*. Here, we will discuss what that means.

A model's purpose is to define a representation for data that mimics the underlying principles that create the data. A model can represent the relationship between GDP per capita and crime or even the logic that goes into how a person drives a car. We can create models for just about anything.

We define a model by defining an *objective function* that the model should minimize or maximize.

Let's take a simple example of two variables x and y that have a linear relationship. We can represent y in terms of x using,

$$f(x) = w_1 x + w_0, \tag{1}$$

where w_1 is the slope of the line and w_0 is the bias or intercept. $f(x)$ can accurately represent y if and only if there in fact exists this linear relationship and x is the only variable that influences y. In reality, this rarely occurs. If we have a linear relationship between x and y, our real $f(x)$ is usually,

$$f(x) = w_1 x + w_0 + \epsilon, \tag{2}$$

where ϵ is a function of all the other variables that can influence y. Creating a function that can accurately represent all possible variables that influence y is infeasible. What we do instead is focus on finding the *best-fit line*. The best fit line, also known as *linear regression*, is a line that minimizes the squared[2] euclidean distance between the line and the actual data points. If no other variables influenced y then we would be able to find a line that minimizes this distance to 0 . We can represent this formulation as,

$$\min_{w_1, w_0} \sqrt{\sum_i^n (y_i - (w_1 x_i + w_0))^2}^2, \tag{3}$$

which, in more words, is minimizing the square of the distance between the actual y and the estimate for y by changing w_1 and w_0.

This formulation can be generalized for a polynomial regression defined by $\mathsf{Y} = \mathsf{w}^T\mathsf{X}$ as,

$$\min_{\mathsf{w}} \; ||\mathsf{y} - \mathsf{w}^T\mathsf{X}||_2^2, \tag{4}$$

where X is an $(m+1) \times n$ matrix, $x_{ij} = x_i^j$, w is an $(m+1) \times 1$ vector, and $||\cdot||_p$ is the ℓ_p-norm[3].

In general, the theory approach in machine learning focuses on formulating new objective functions and finding solutions to them. In other words,

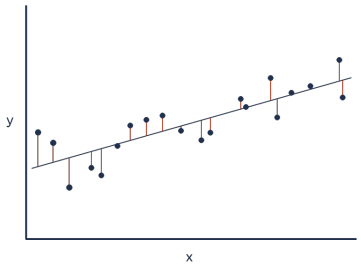

Figure 2: Example of a best fit line with highlighted distances between each point and the line.

[2] Why squared? We use the squared euclidean distance because we can easily derive a closed-form solution to that objective.

The following

$$\min_x f(x),$$

reads as, minimize the value of $f(x)$ by changing the value of x or find the value of x that determines the minimum value of $f(x)$

Variable	Meaning
x, α	Scalar value
x, v	vector value
X, A	matrix value

Table 1: Variable Notation

[3] The ℓ_p-norm is defined as

$$||\mathsf{x}||_p = \sqrt[p]{\sum_i^n x_i^p}$$

, where $p \geq 1$ for example:

$$||\mathsf{x}||_1 = \sum_i^n x_i$$

$$||\mathsf{x}||_2 = \sqrt{\sum_i^n x_i^2}$$

developing new models and determining methods to accurately estimate their parameters.

Practitioners

Machine learning practitioners start with a variety of available models that they want to use on a particular dataset. First, let's define *data*. Data is any information we collect about an object to represent or explain it. Examples of data include; the number of sales per month of every product at a store, the audio of a phone call, the text in an email, a photo of a particular object, or the genetic structure of an organism. With machine learning, we want to create a model that represents the data. Representing the data means that the model can predict how one value changes when compared to others (supervised), detect a pattern or complete the data (unsupervised), or make decisions about the data (reinforcement). This will become clearer as we go over examples of these models.

Metadata is data about data. You can think of data as the contents and metadata as the description of that. Examples: audio of a phone call is data, the two numbers on the call and the time of the call are metadata, text in an email is data and the sender, recipients and time of email are metadata.

Model Fitting

The general model fitting procedure is that we split all the available data into a training set and a test set, then we train our model on the train set and test it on the test set.

When we train a model, we need to understand its effectiveness to compare it with other models and determine the best one. Different problems will have different metrics that we use to measure effectiveness but regardless of the metric chosen, the model should generalize and not overfit or underfit.

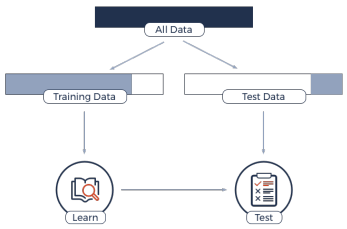

Figure 3: General model fitting procedure

- Overfitting - training a model that performs well on training data but performs poorly on test data.
- Generalizing - training a model that learns enough from the training data to generalize a pattern that performs well on test data
- Underfitting - training a model that performs poorly on both training and test data.

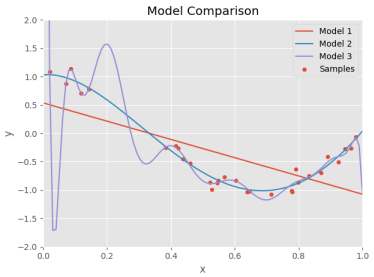

Figure 4: An example of 3 models fitting sample data. Model 1 underfits the data, Model 2 generalizes and Model 3 overfits the data.

Every model has parameters that it learns through training, ex: coefficients of a polynomial. Some models also have *hyperparameters* that are inputs to the model training, ex: the order of the polynomial to use. In example in figure 2 we see that changing the polynomial order leads to varying models that underfit or overfit. With a simple example we can visualize the models and determine the best one but with more complex methods it may be difficult to do so. *Cross-validation* is a method used to determine the ideal hyperparameters of a model.

After splitting the data into train and test sets, we split the train set into model and validation sets. We then train our models on the model set using

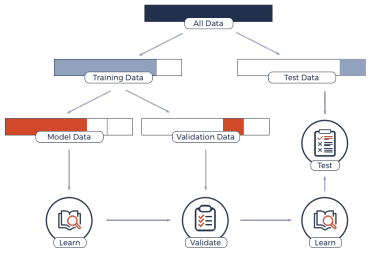

Figure 5: Model fitting using cross-validation.

different hyperparameters that we validate on the validation set. **Once we determine the best hyperparameters, we train a new model using them on the entire train set**. We can determine the best hyperparameters by finding the highest cross validation score. An easy way to do this is to plot the scores for both the training set and validation sets versus the hyperparameters.

We mentioned that we will split the data into train and test and then split train into model and validation but we did not discuss how we do so. We usually use a random selection to split the data after shuffling it. To be even more careful with understanding how well our model performs, we can use k-fold cross validation. k-fold cross validation is the process of breaking up our train data into k equal portions and repeating the cross validation experiment k times. For each experiment we select one of the k portions as validation and train the model on the remaining portions.

Once we find the best model, the task is complete and we can begin using that model to make predictions or whatever else its role may be.

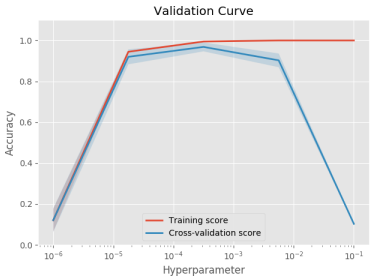

Figure 6: An example validation curve used to identify the best hyperparameters.

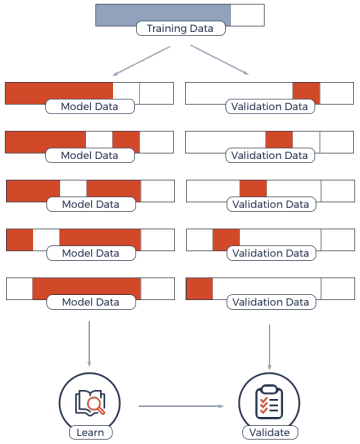

Figure 7: An example of k-fold cross validation with $k = 5$.

Product

With a product approach, the key focus is on a measurable value produced for the customer or client. Machine learning is simply a tool that can assist in the production of that value. A product often does not need the best model possible to create value. With the product approach, the focus is twofold, business and engineering.

The business aspect focuses on determining where machine learning can produce added value. It also requires domain knowledge and a general understanding of the capabilities of machine learning. After completing this course, you should have the machine learning portion of that covered. As important as it is to know what machine learning is capable of, it is even more important to understand what it is **not** capable of.

The engineering aspect focuses on determining how machine learning plugs in to the product development pipeline from data access to prediction delivery. It's important to understand how to get access to data and how to deliver predictions. Later on in the course we will go over code examples of accessing data. As for prediction delivery, automated tools exist that can convert a trained machine learning model into a web-based API that can interact with front end services such as a mobile app or web application. Even if you don't use automated tools, simple frameworks such as Flask for Python and Express for JS can get you building an API fairly quickly. This greatly simplifies the prediction delivery process from models.

This course will focus on the Theory and Practice approaches. Throughout the course we will discuss applications of machine learning and how those can fit within products.

For the semester project, a product is highly encouraged. Product building requires knowledge outside of the scope of this class such as web or app development and domain expertise. External resources are listed elsewhere for those interested in a product focus for the course.