Section 1: Week 3: Applying Machine Learning

Nate Bachmeier

TIM-8130: Data Mining

February 22nd, 2020

North Central University

Applying Machine Learning

# Section I: Finding Correlations

# Section II: Application of Machine Learning to Organizations

Many organizations are collecting vast quantities of data with the hopes of unlocking the potential by transforming it into business intelligence. A common misconception arises when these businesses assume that adding machine learning will magically create insights. In reality, data cleaning and curation accounts for roughly 70% of the effort (Gibert, Sanchez-Marre, & Izquierdo, 2016) and meta-learning analysis accounting for the remainder (Tripathy & Panda, 2017). The No Free Lunch Theorem explains the disconnect by stating that no universal algorithm exists that is equally efficient for all scenarios. That statement makes sense, given that a machine learning algorithm produces a scenario-specific statistical model. For instance, determining association rules requires Apriori-like algorithms that focus on groupings of frequent itemsets. Both the approach and end goal are significantly different than logistical regression, which maps features to a continuous range. Even similar objective strategies, such as classification versus clustering, are dissimilar because of the unique mapping characteristics between labeled and unlabeled buckets. These distinctions suggest that a successful application of machine learning requires (1) a concise business question, (2) relevant facts about the issue, and (3) an algorithm that transforms the evidence into answers.

Having a specific business question is the first and most critical step in the process. Without a clear objective, the project is unlikely to be successful. Consider the impact of attempting to discover cures to heart diseases with banana cells. The engineering team is immediately at a disadvantage against a competitor that is using heart cells. As the fruit cell data set continues refinement, these facts remain irrelevant to their goal. No matter the algorithm, it is improbable that a revelation arises. However, if the specific question asks for similarities between fruit cell and heart cell decay, then these facts are now relevant. Other examples are less black and white, such as predicting website click-streams or product recommendations. These scenarios can be more nebulous, and only through a refinement of the target objective does it become clear whether evidence is significant.

After determining that a collection of facts address the business question, the next step is to curate them into an efficient format for analysis. Typically the evidence will come from multiple sources such as customer reviews, order histories, search histories, and third party providers. The heterogeneous sources use different identifiers, categorical labels, and value scales (e.g., inches versus centimeters) that need to become standardized before processing the results. Along with standardization, the data sets need to be joined, aggregated, deduplicated, and filtered for erroneous examples. When these actions do not take place, then the statistical predictions will be skewed and contain bias. Some of these incorrect values are easy to detect, such as a value that is several deviations from the mean. Other scenarios, like miscalibration, are more difficult to detect because they are still within statistical norms. As the volume of data increases, so will the frequencies of these challenges. While it is economically prohibitive to detect and correct every instance, a best-effort needs to occur to avoid a “garbage-in/garbage-out” scenario. Though at the same time, it is critical to understand why the removal of examples from the training sets is necessary. For instance, a list of security transactions could enumerate the date that a customer purchased and sold the stock. If an empty sales date represents the asset is still held, then blinding dropping rows with any empty column would discard the entire user-scenario. Depending on the parameters of the question, this invalid action could result in a model that provides excessive confidence through overfitting and does not make reliable predictions in practice.

With our curated dataset in hand, the data engineers are ready to begin applying various machine learning algorithms. It is tempting to start with highly sophisticated strategies that leverage state-of-the-art neural networks that feed into some other monstrosity, though a better approach starts with dumb algorithms and adds complexity only when needed (Witten, 2011). Many datasets contain combinations of dominant, irrelevant, and misleading features that require attention upfront. Consider a feature set that describes the characteristics of children, and the researcher is attempting to predict their current school year (e.g., 1st versus 3rd grade). In this scenario, using only the age attribute is more accurate than counting their teeth or considering height. Witten (2011) provides a collection of strategies, such as one-rule decision trees, that programmatically surface these dominant features, which then form the foundation of more complex models. Perhaps the researcher is interested in student assessment results and wants to recommend additional learning materials. An efficient starting point is to use k-mean clustering to partition the records based on similarity. Then instead of investigating all instances within the dataset, representatives from each partition can be selected to derive insight about the broader population. Alternatively, the analyst might use the assessments to predict future scores using logistic regression to approximates a curve that explains the parametric relationship between a feature set and target value. Often the feature set is nonparametric, and this introduces the need for an intermediate approximation of the parametric equation (Brown & White, 2017). Multi-Layer Perceptron (MLP) accomplishes this feat by using backpropagation to estimate edge weights in a connected graph between the feature nodes and the “hidden layer.” This process can recurse multiple times until the nonparametric equation maps to a parametric equation that accurately predicts the target value. One of the challenges with this strategy is that it can become overfitted, especially in the presence of sparse data. Apriori is an effective strategy to mine association rules, that does not require extensive knowledge about the dataset shape. The procedure begins by finding the most frequent single items and then selecting those that occur more than a threshold. This subset becomes a filter when choosing the most common two-item sets, and that recurses until no more combinations have sufficient support. After discovering common subsets of attributes, the analyst can segment the population into different clusters.

Specific scenarios can require more sophisticated algorithms to make predictions about the dataset. For instance, time-series information, natural language processing, autonomous vehicles, and video processing do not naturally align with the previously discussed solutions. Research into these domains requires deep learning through neural network technologies (Keras.io, 2020). Similar to MLP, deep learning creates connected graphs and then estimates the edge weights to map nonparametric feature sets to parametric results. Keras is a high-level abstraction layer that allows data engineers to prototype these technologies rapidly. A key strength of this library comes from its focus on smart defaults and consistent interfaces so that users do not need advanced mathematical degrees. It includes numerous algorithms, such as applying Long-Short Term Memory (LSTM) to sequential structures and Convolutional Neural Networks (CNN) for image analysis. Even though these algorithms are more complex, their application follows the same pattern as SciKit-Learn and similar technologies. The process begins with reshaping the curated dataset, then calling the fit method, followed by the evaluation method. In SciKit, there is a concept of pipelining, which allows for several pre-processing actions to become chained together. For instance, the pipeline might scale the dataset and then reduce the dimensionality through Principal Component Analysis (PCA), to improve training results. Keras offers a similar pattern by allowing the outputs of an algorithm to flow directly into another. These nested layers enable scenarios such as begin with an LSTM to inject noise (e.g., Gaussian), feature boosting (e.g., softmax), feature reduction (e.g., MaxPool), and even transition into strategies such as multi-classification. When using either simple statistical models or complex deep learning, it is essential to follow scientifically sound patterns. For instance, injecting too much noise or reducing features with an invalid filter will result in erroneous values.