Section 2: Week 4: Techniques to Discover Logic

Nate Bachmeier

TIM-8130: Data Mining

March 1, 2020

North Central University

Techniques to Discover Logic

# Section I: Determining Relevance

## Identify Business Value

It is not possible to answer a question if either the question or the necessary facts are not known. Consider the scenario where the organization wants to execute the most efficient marketing campaign using the least amount of resources. Without proper planning, the business might stumble upon an acceptable deliverable (local maxima) though they are unlikely to encounter the global maxima. If instead, the company specifically defined the objective as increase awareness of their product to minorities and underserved rural populations, then it becomes possible to rate the quality of supporting evidence. Now that a logical base case exists, the company can review public and private data providers and perform an initial inclusion filter. For instance, governmental census information contains population statistics that describe high-value segments to place physical advertisements. Though even within this vast dataset, only a subset will be useful today or even tomorrow. Perhaps the organization has a strong presence in the southwest, and the business model does not support expanding into the northeast (e.g., licensing, political, or transportation concerns). These limitations remove the need to have either humans or machines mine those areas (partitions), as there is no potential value.

## Identify Completeness

After repeating this filtration process multiple times, the conversation can transition the focus to the shape and volume of these facts. If the analyst is attempting to build a statistical model across five thousand features, they will need a lot more data than an alternative that contains five features. Waal and Toit (2011) suggest that a minimum of ten examples need to exist for every parameter. That logically makes sense as a deficit of information leads to speculation and bias. Imagine asking two random people at an NYC Metro bus stop their income, and then predicting the median for the region. Alternatively, a person that samples from different neighborhoods along the Metro is more likely to estimate this value accurately. Hsu et al. (2017) caution that the absence of evidence is not evidence of absence, and that under-generalizations frequently occur. For instance, if the collection of these samples takes place during the afternoon, it is likely to miss high earning professionals that commute during the mornings and evenings. Data miners can detect some of these gaps by looking at descriptive statistical and broad aggregates. Consider how a pivot against time or industry (e.g., financial, technology, hospitality, etc.) would highlight missing examples from the metro scenario. When the shape of these pivots does not align with expectations, then there is a problem.

## Identify Shape

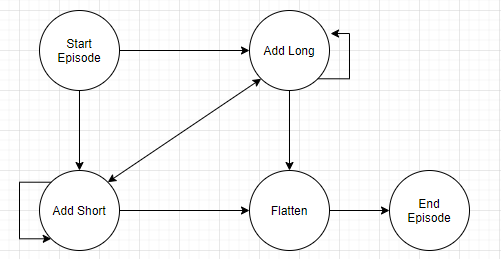
Understanding the business-value and determining a complete collection of data enables the organization to begin testing their hypothesis. Snee (2015) directs research towards a graphical investigation first. For instance, plotting a simple line chart of market segment growth over time makes specific cyclical trends visually discoverable, versus a detailed analysis of rows in a table. Another critical strength of graphically approaching the data's shape comes with the relative ease of explaining rational to other parts of the institution. Not all team members are 'deeply in the trenches' and need some mechanism to connect the discoveries with their external perspectives. Consider the distinction between presenting *only* a polynomial equation to predict housing prices versus the inclusion of a scatter plot and estimated curve. Aside from data visualizations, the input shape is an important characteristic of mining algorithms. If the algorithm learns on individual instances, then aggregate counts are not useful regardless of its relevance to the problem. Similar requirements are present concerning time components and multi-dimensional data (e.g., images). These constraints can force the researchers to search for more supporting evidence or alternative expressions of the problem. However, specific scenarios are inherently spare, or it's prohibitively expensive to acquire more data, such as collecting project bids from contractors. Instead of mapping these bids to precise quantitative estimates through regression (e.g., $152,593.22)—the company could use a qualitative bucketing strategy (e.g., cheap versus expensive). While the qualitative approaches are less specific, they provide a mechanism to reduce the search space further and can ensemble with other algorithms to form a unified collection of signals.

# Section II: Compare Predictive Techniques

## Markov Chains

A core challenge to applying basic statistics toward real-world data comes from the assumption that each action is independent. However, many scenarios contain a conditional state transition probability that is dependent on the current state. If the stock market falls 5%, should an investor buy more stocks? The binary question requires a contextually sensitive answer that considers their net position (e.g., short the market), outlook (e.g., 2008 financial crisis versus 2017 Trump bump), and similar factors. Markov chains provide the mathematical basis for making statistical models that incorporate these dependencies (Kahn Academy, 2014). Creating the hypothetical purchasing model (see Figure 1) begins with a state diagram that represents the different actions available. Then Monte Carlo solutions can approximate each edge's weight by random sampling and recording the decisions. While multiple use-cases can follow the same model, the weights are scenario-specific, such as (a) 401k retirement account that only adds index funds versus (b) delta-neutral (directionless) options trader. This trait is similar to other algorithms where efficient training requires highly relevant facts to specific questions.

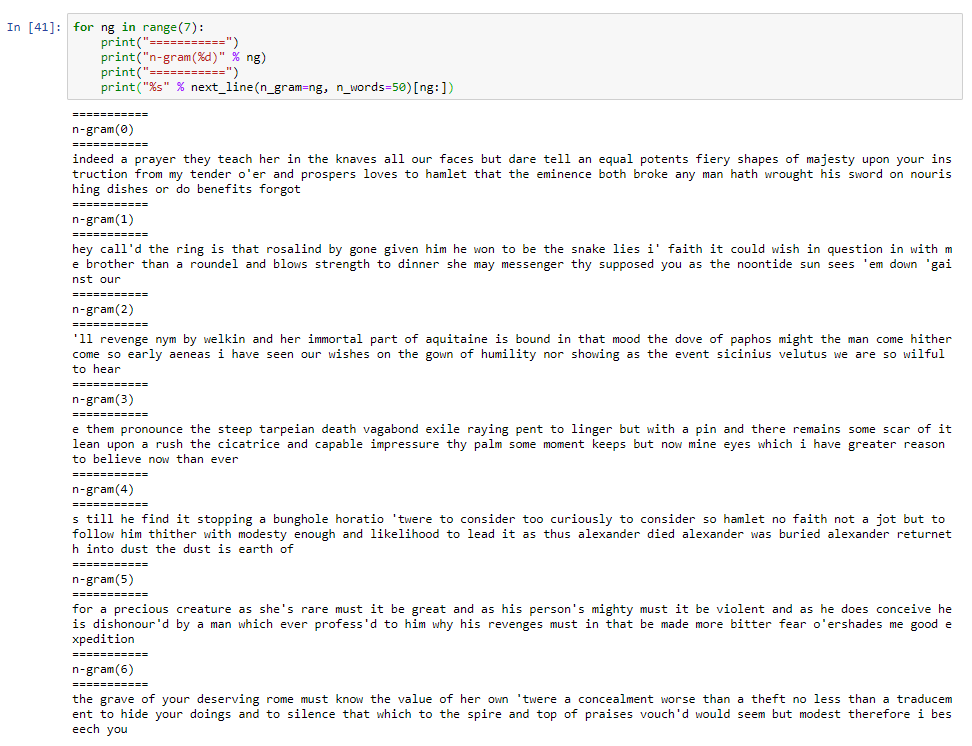
Figure 1: Purchasing Model



## Markov Experiment

Many online tutorials recommend exploring Markov chains as a solution to predict the next token in a sequence. Mason (2003) maintains an open-source repository of Shakespeare plays, which is easy to mine for different related expressions. An experiment began with downloading each play and normalizing the text into a corpus of lowercase words. Next, an iterator constructs a word\_dictionary that maps n-gram tuples to a word bag of immediately following values. Then traversal of the Markov model chooses a random starting point, then selects a random next word, iterating until a stop condition. Across the test iterations, tests of different n-gram sizes (degrees of freedom) ranged from one to six. The higher the count, the more natural the sentences sound, largely due to overfitting. Even at low n-gram terms, a frequent challenge arose from many unique words that cause long sequences of static choices.

Figure n-gram Examples

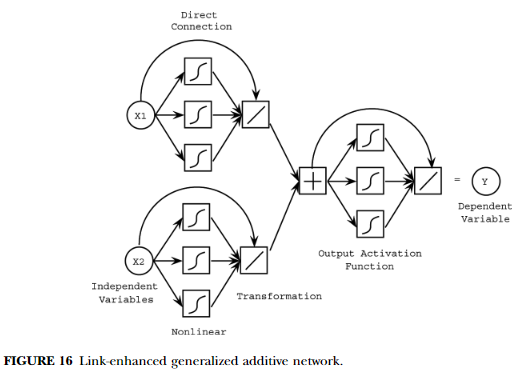


## Neural Networks

The goal of a Multi-Layer Perceptron (MLP) algorithm is to map a non-parametric set of inputs to a parametric set of outputs, by approximating an intermediary mapping function (the hidden layer). A fully connected graph can represent this structure, such that all inputs connect to the hidden layer, which in turn connects to all outputs. Then through an iterative process, examples are fed through the graph, followed by backpropagation adjusting the weights in response to the chosen output versus the expected value (Ng, 2016). According to Fridman (2017), backpropagation is a recursive process of taking the partial derivative of two logic gates and then applying a weighted update. He expands on the idea of these connected graphs with an example of image classification passing through several three layers to extract edges, corners, object parts, and finally, predict object identity. While the mathematical basis and engineering steps are fairly procedural, the efficient design of the network architecture requires both art and science.

Perhaps the artfulness comes from a lack of planning or awareness of how the *ensemble* of distinct training subsystems combines. There's no reason to assume every node is fully connected or has an edge weight above zero (see Figure 3). A logical representation might consider feature 'x1' connected to N neurons that regress one output, with feature 'x2' implementing some classification pattern. These network segments are producing signals that collaborate to provide a richer inference about the broader topology. It would, therefore, stand to reason these network segment microstructures extrapolate and continue to be present in larger and more complex processing networks. The solutions by both BellKor (2007) and Li et al. (2019), suggesting this assumption is generally accurate.

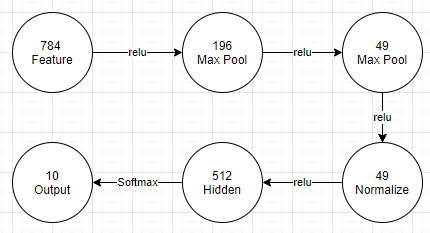
Figure GANN Architecture (Waal & Toit, 2011, p. 399)



## Neural Network Experiment

Consider the scenario of mapping 28x28 images of clothing to ten categorical labels (e.g., hats versus coats). The number of input features (neurons) is 784, and there will be ten output neurons—how many neurons should exist in the middle? Rosebrock (2019) provides an example solution (see Figure 4) to Fashion MNIST that begins with feature reduction through two max-pooling hidden layers and batch normalization. After cleaning, the solution uses a single 512-neuron hidden layer to predict one of ten output categories (with softmax). Reducing the size of the hidden layer to 128 or 256 has minimal impact on the cross-validation scores, though really low values of 5 to 16 negatively impact accuracy. In this specific example, changing the activation functions (e.g., softmax to tan-h) creates more performance fluctuation than any other knob, with model accuracy ranging from 20 to 85%.

Figure : Fashion MNIST



# References

Bell, Koren, & Volinsky. (2007). The BellKor solution to the Netflix Prize.

Fridman, L. (2017, January 16). *MIT 6.S094: Introduction to Deep Learning and Self-Driving Cars*. Retrieved from YouTube: https://www.youtube.com/watch?v=1L0TKZQcUtA&feature=youtu.be

Kahn Academy. (2014). *Origin of Markov Chain*. Retrieved from Kahn Academy: https://www.khanacademy.org/computing/computer-science/informationtheory/moderninfotheory/v/markov\_chains

Li, Zhao, Zhang, & Chen. (2019). Reinforcement Learning and Deep Learning Based Lateral Control for Autonomous Driving. *IEEE Computational Intelligence Magazine May*, 83-98.

Mason, G. (2020, March 1). *Shakespeare's plays*. Retrieved from OpenSource Shakespeare: https://www.opensourceshakespeare.org/views/plays/plays.php

Ng, A. (2016). *Machine Learning*. Retrieved from Coursera: https://www.coursera.org/learn/machine-learning

Rosebrock, A. (2019, February 11). *Fashion MNIST with Keras and Deep Learning*. Retrieved from PyImageSearch: https://www.pyimagesearch.com/2019/02/11/fashion-mnist-with-keras-and-deep-learning/

Snee, R. (2015). Practical Approach to Data Mining: I Have All These Data; Now What Should I Do? *Quality Engineering, Volume 27*, 477-487.

Waal, d., & Toit, d. (2011). AUTOMATION OF GENERALIZED ADDITIVE NEURAL NETWORKS FOR PREDICTIVE DATA MINING. *Applied Artificial Intelligence. May/Jun2011, Vol. 25, Issue 5*, 380-425.