DisputeSoft
Software Dispute Experts

An Overview of Machine Learning Applications to Technology Assisted Review for E-Discovery Part One: Supervised and Unsupervised Machine Learning Algorithms

Sarah Bullard, formerly of DisputeSoft

This is the first article in a four-part series on technology assisted review (TAR), a process that uses machine learning to increase efficiency and decrease the cost of document review in litigation.

This article explores differences between supervised and unsupervised machine learning algorithms, and explains why supervised algorithms are a better fit for TAR in e-discovery.

Supervised Learning Algorithms

Most TAR machine learning algorithms fall under the category of supervised learning. Supervised learning involves human provision of a training data set to the algorithm. A training set (or "seed set," as it is typically called in TAR for e-discovery) contains data points that have already been classified by a qualified human being. For e-discovery, this usually involves an attorney classifying documents as "responsive" (relevant) or "unresponsive" (irrelevant). The criteria for categorizing a given document as responsive or unresponsive depend greatly on the nature of the matter itself. For example, in a matter concerning illegal gambling, the legal team will mainly want to examine documents that pertain to gambling in some way. So a document that mentions "gambling" or terms related to gambling should be classified as responsive, while one that doesn't should be classified as unresponsive.

After the testing set is classified by the system, a qualified human reviews the results and decides whether more training is necessary before allowing the system to classify the rest of the documents.

For best results, the seed set should include a diverse selection of documents representing all issues of interest to the matter. The system will then use its algorithm of choice to come up with a function that is designed to make predictions about the matter based on the seed set. It then applies that function to a second set of documents (known as a "testing set") passed in without classifications. After the testing set is classified by the system, a qualified human reviews the results and decides whether more training is necessary before allowing the system to classify the rest of the documents.

Supervised learning can be performed with a variety of algorithms, which may incorporate regression and decision trees.

Regression Algorithm

Machine learning algorithms work with numbers and data points, not with words. Yet the documents in need of analysis are made up entirely of words. So how can they be made suitable for machine learning? Fortunately, algorithms exist to turn each document into a data point in multidimensional space. One such algorithm counts the number of occurrences of each term across all documents in the seed set. Another algorithm calculates the importance of each term within the context of all documents in the seed set based on how frequently it occurs compared to other terms. This process of turning each document into a data point is called vectorization. To learn more about vectorization algorithms, review this article from Analytics Vidhya.

Once documents have been converted from words into data points, they must be classified as responsive or unresponsive. For this, we need a classification function. This is a function that takes a document's vectorized data point as an input and determines whether it is responsive. There are different types of classification functions that can be applied to e-discovery problems, but they will all examine the frequencies (or frequency-related values) of certain words in a document to determine whether that document is responsive or not.

First, either an algorithm or a data scientist will choose the type of classification function to use for the document classification problem. Once the type has been decided, however, there are still

important choices to make. There are a vast number of functions of each type. So, which one should be chosen for a specific document classification problem? Fortunately, other algorithms exist to choose the right function for the problem in question. The most popular of these algorithms is called gradient descent. To learn more about gradient descent, follow this link to a helpful explanation from KDnuggets.

Once gradient descent or a similar algorithm has chosen the best possible classification function it can find for the given seed set, it is time to test out that function. Data scientists will provide that function with a testing set as an input. The testing set is similar to the seed set in that it is large enough and diverse enough to accurately represent the entire available body of documents. The testing set will also be classified by an expert, and then fed into the chosen TAR function, this time without the classifications. The function will make its prediction about each document in the testing set, and then the data scientist will compare its predictions to the classifications provided by the expert.

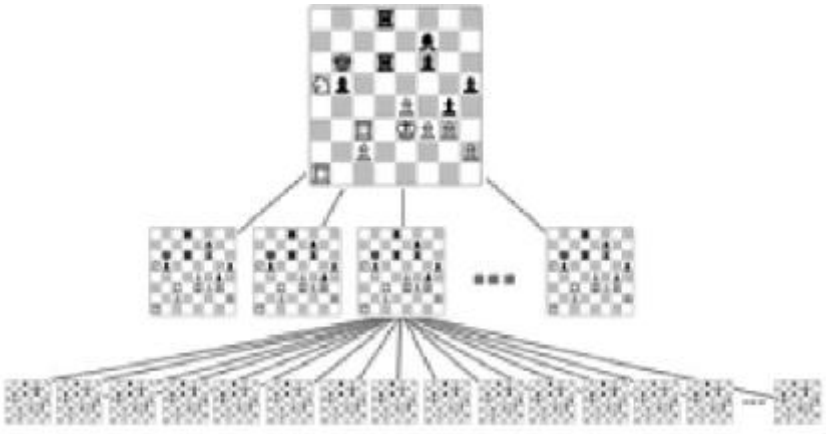
If the function's predictions meet or exceed a predetermined threshold of accuracy when compared to the expert's classifications, the data scientist will allow the classification function to classify the rest of the documents that remain. If the function's predictions do not meet or exceed such a threshold, the data scientist must continue to work to improve the function. They may choose a larger or more diverse seed set with which to train the system. They may choose a new type of classification function from which the gradient descent algorithm must choose. Or, they may start the gradient descent algorithm at a different data point than the one at which it started the first time, since its starting point can affect its decision. These steps can be repeated until the gradient descent algorithm chooses a classification function that performs adequately on the testing set, at which point the chosen function can be trusted to classify the rest of the documents.

Decision Tree

A decision tree offers sets of choices to the algorithm, much like a choose-your-own-adventure book. The algorithm makes these choices based on characteristics of the data point in question. It works its way down the tree until it reaches a "leaf node," a point where there are no more decisions offered. The leaf node contains the final classification for that data point. It is

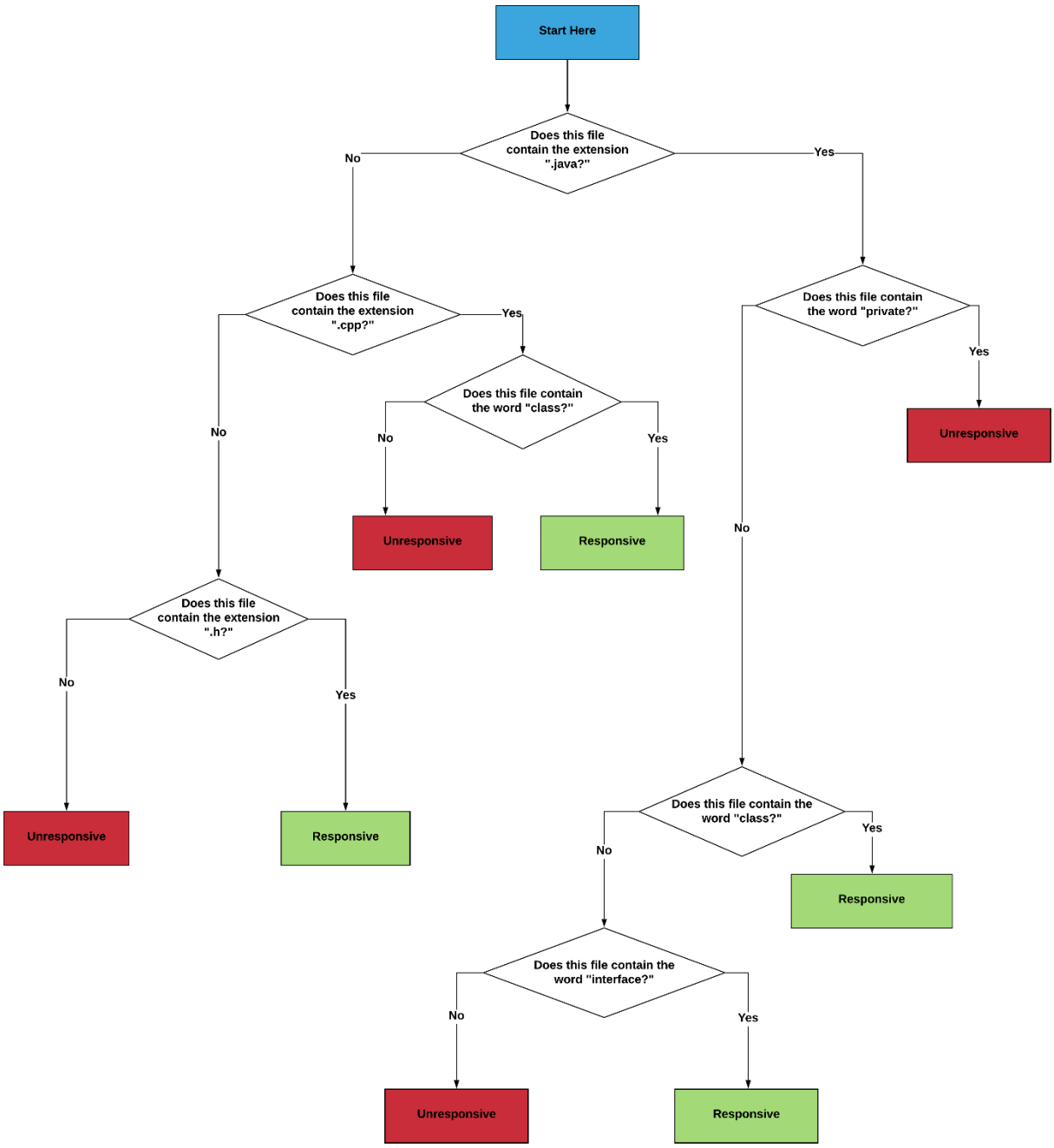
analogous to the ending of a story in a choose-your-own-adventure book, at which point all of the reader's choices have culminated in an unchangeable result.

For example, IBM's original chess machine, Deep Blue, was coded with a staggeringly enormous decision tree containing all possible outcomes of all possible moves. As moves were performed, it used a pruning algorithm to remove the outcomes that were eliminated.



A fragment of Deep Blue's decision tree. Each move is a decision that changes the decisions that can be made in the future. Image taken from Stanford's CS221 webpage Deep Blue.

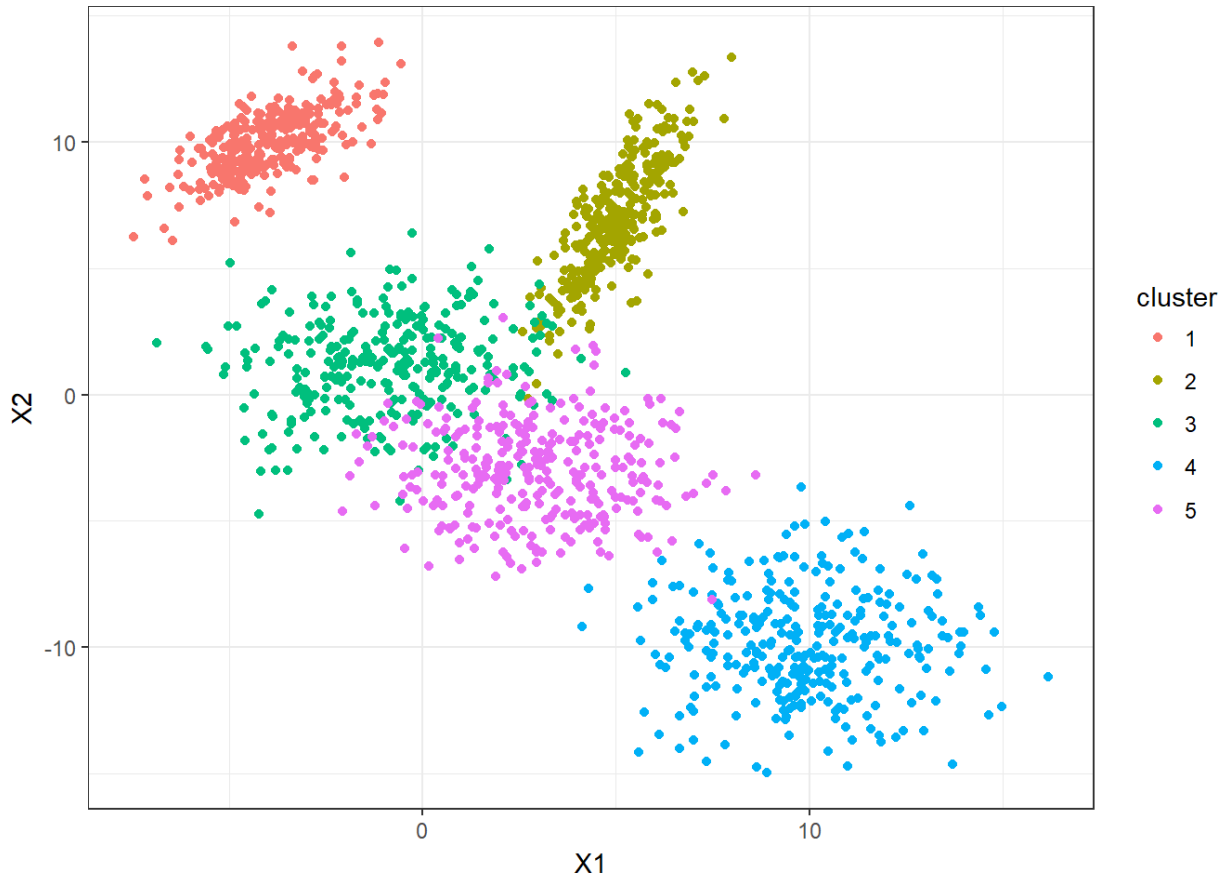
Consider an e-discovery case in which the decision tree algorithm has learned that all responsive documents are either Java source code files, C++ source code files, or C++ header files. Among Java source code files, only those defining public, protected, or package-private classes or interfaces have been determined to be responsive (i.e., Java source code files defining private classes or fields have been determined to be unresponsive). Additionally, among C++ source code files, only those that define classes have been determined to be responsive. An example of a decision tree that might result from this learning process is shown below:



Do these trees have to be programmed by hand, each question carefully considered and coded directly into the algorithm? Fortunately, no! Several algorithms can generate a decision tree given a set of pre-classified training data. These algorithms take the attributes of the data points into consideration and determine which questions are the most effective to ask at which points in the tree.

Unsupervised Learning Algorithms

Machine learning algorithms that are not trained with a pre-classified seed set are known as unsupervised learning algorithms. Unsupervised machine learning involves finding groups of data that are similar to each other according to the algorithm's perception rather than sorting them into predetermined categories. If you have an iPhone, you might have noticed your Photos app using unsupervised learning to categorize your pictures according to their human subjects. Even though you never have to tell the app which face belongs to which person, the phone is able to group photos that feature the same people based on which faces are similar to each other and different from others. It doesn't need user input to make these distinctions.



The results of running a k-means clustering algorithm, which is a commonly-used unsupervised learning algorithm. The machine has grouped the points into different color-coded, unnamed categories based on their proximity to each other. This image is taken from Thean Lim's explanation of k-means clustering.

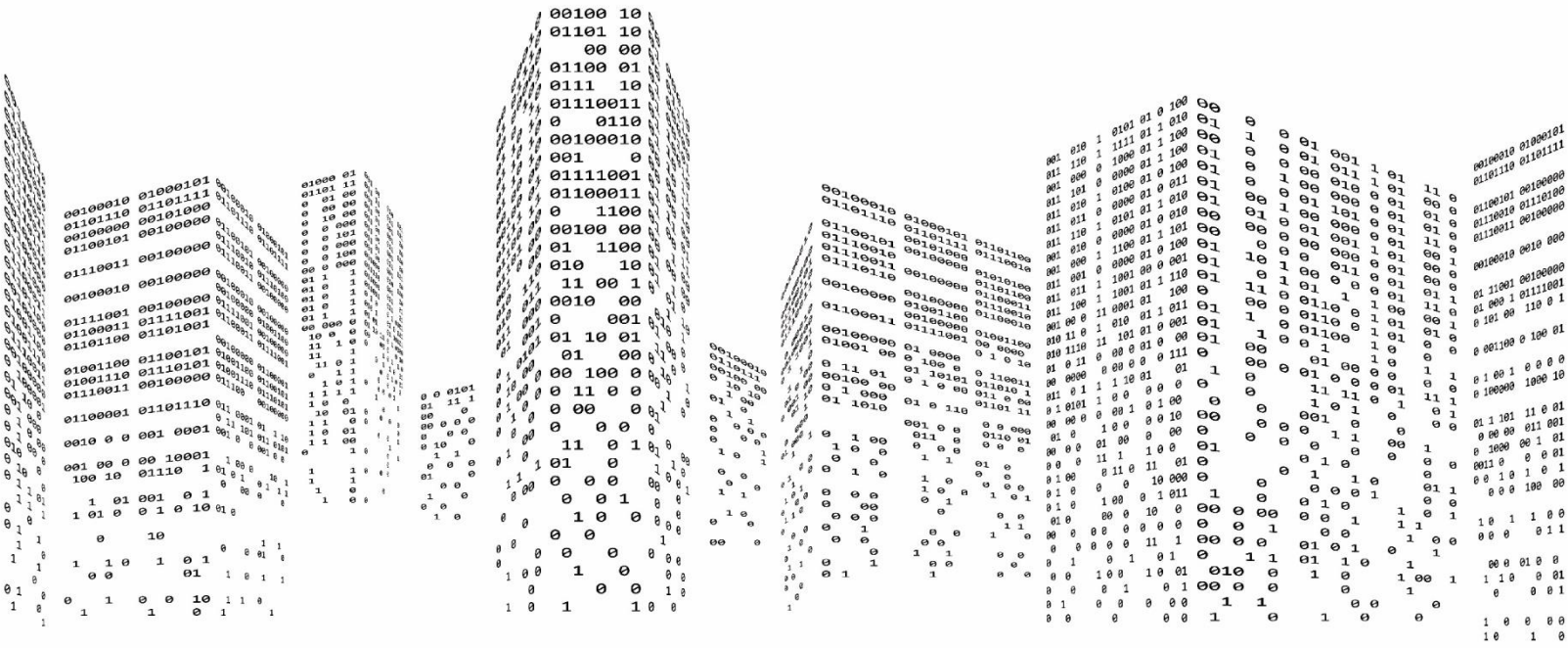
Unsupervised learning gets more interesting and useful when the machine chooses groupings that a human might not choose. In the figure above, you'll notice that the machine categorized some data points as pink that might look like they belong in the green area. This could be a mistake, but it could also be the most effective grouping and lead to new insights about the data in question.

Unsupervised machine learning is effective for discovering new connections in datasets, but typically is not used for TAR, since attorneys generally wish to identify specific types of documents that fit into a set of predetermined categories when performing e-discovery.

Conclusion

Attorneys litigating software-related matters must often review large volumes of electronically stored information (ESI) potentially relevant to a dispute. Technology assisted review of such documents can significantly reduce e-discovery time and costs. Although supervised algorithms are the standard choice for e-discovery, both supervised and unsupervised algorithms are used for different types of machine learning applications. To learn more about the role of machine learning applications in TAR for e-discovery, check out the next article in this series, which explores two other applications of machine learning: optical character recognition and natural language processing.

DisputeSoft's approach to e-discovery goes beyond traditional document review to include identification, recovery, preservation, and analysis of systems, databases, and other non-custodial evidence. Learn more about DisputeSoft's electronic discovery services and explore a representative e-discovery case: General Electric v. Mitsubishi Heavy Industries.



If you are in need of an e-discovery expert, we invite you to consider DisputeSoft.

Contact Information

Jeff Parmet, Managing Partner

301.251.6182

jparmet@disputesoft.com

12505 Park Potomac Ave. | Suite 475 | Potomac, MD | 20854