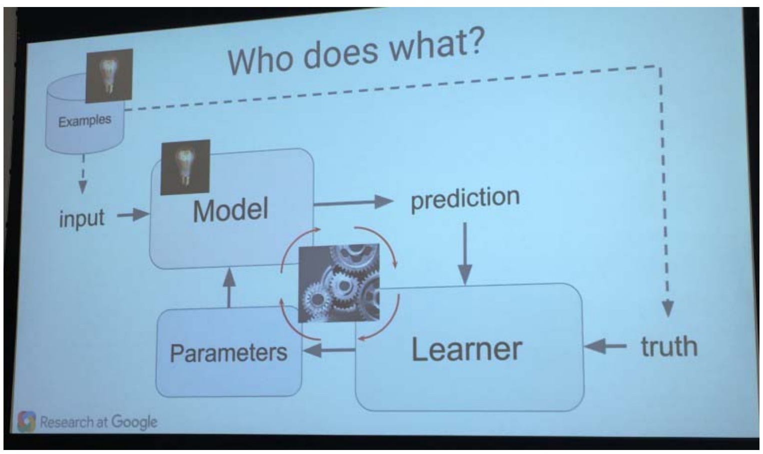
Automatic and Transparent Machine Learning

Are you a citizen data scientist or do you plan to become one? If so, then you have quite an adventure ahead of you. Like all your data science peers, you will also discover a few challenges as you begin to practice your art. Two of the foremost difficulties involve creating accurate machine learning models, and then being able to explain those models' prediction results. These are the two big issues we take on in this article, as we present two recent advancements that make the machine learning process automatic and transparent.

But first, what exactly is a machine learning model? Simply stated, a model is an algorithmic construct that produces an output when given an input. A model is the product of training a machine learning algorithm on a training dataset.



Caption: A high-level view of a typical machine learning architecture. Image courtesy of Google.

The training data are the key, as they contain the "right answers" that the machine learning model will reference as it makes its predictions when presented with new data. Now here is the rub: building an accurate model is a time-consuming, iterative process involving a rather tedious sequence of steps. The folks at Amazon Web Services summarize these steps as follows:

1. Frame the core machine learning problem in terms of what is observed and what answer one wants the model to predict.
2. Collect, clean, and prepare data to make them suitable for consumption by model training algorithms. Visualize and analyze the data to conduct sanity checks of data quality and to understand the data.
3. Often, the raw data (input variables) are not represented in a way that can be used to train an accurate model. In this case, one should try to construct more predictive input representations or features from the raw input variables.
4. Test many model configurations iteratively. Each configuration involves a machine learning algorithm and a specific setting of its tuning knobs. For each configuration, feed the features and training data to the learning algorithm to train a model. Evaluate model quality on data held out from model training. Select the best model configuration from the many tested ones. The corresponding model becomes the final model.

5. Use the final model to make predictions on new data instances.

Step 4 involves model selection, a fundamental task of scientific inquiry. Not all model configurations are created equal. Given the myriad mechanisms and processes of data generation, how can one select a good model configuration that can result in accurate predictions on incoming new data?

As it stands today, comparing different algorithms and settings of their many tuning "knobs" is a trial and error process. The knobs one tunes on a machine learning algorithm are called "hyperparameters." Let us put this in the context of a deep neural network. The network designer must make many decisions on hyperparameter values prior to model training. In a convolutional neural network, examples of such decisions include: for each convolutional layer, how many features will be used and how many pixels are contained in each feature? For each pooling layer, what window size and stride will the model use in traversing an input image? For each layer type, e.g., pooling layer, how many layers of this type will the model include? In what order will the layers be arranged? Keep in mind that some networks can have hundreds or even more than one thousand layers.

Other examples of hyperparameters include the choice of kernel used in a support vector machine and the number of neighbors k in a k-nearest neighbor classifier. With several dozen commonly used machine learning algorithms, so many hyperparameters, and numerous possible values of those hyperparameters, one can reasonably expect to test only a small fraction of all possible configurations. In other words, unless you are very lucky, you will likely be settling for less than the best possible model configuration and accuracy. What is more, each time you train an algorithm on a training dataset with different settings of hyperparameter values, you obtain a different model. Moreover, good combinations of algorithms and hyperparameter values vary by the specific modeling problem and are unknown beforehand. The combination must be specified before model training starts, and needs to be iteratively refined to find one producing an accurate model. Indeed, hyperparameters control many aspects of an algorithm's behavior and also impact use of computing resources.

The Model Building Process

Given a specific modeling problem, the user first selects a machine learning algorithm and sets a value for each of its hyperparameters. Then the user trains the model on a training dataset. The model's accuracy will likely be low at this stage. This brings us to an important question: what kind of error rate can your application tolerate? For most applications, accuracy matters a great deal.

To improve results, the user changes the algorithm or its hyperparameter values and re-trains the model. The user can expect to repeat this process for several hundred or even several thousand iterations to obtain a model with satisfactory accuracy.

By now, you should have gotten the idea that model selection is a big problem. The selection process is not only labor intensive, but also requires a level of machine learning expertise generally beyond that of a lay user. In fact, the selection process can be difficult even for machine learning experts. So what to do?

Efficiently Automating Machine Learning Model Selection

To overcome this difficulty, we recently developed an automatic method to quickly find a good combination of a machine learning algorithm, feature selection technique, and hyperparameter values for a given modeling problem when many algorithms and techniques are considered.

Model selection is fundamentally a search problem. Our key idea is to use system techniques like sampling and caching to improve search efficiency. We notice that it is time consuming to test a combination on the whole training dataset. In one example, it took two days on a modern computer to train an award-winning model just once on 9,948 patients with 133 features. Yet, having *rough estimates* of multiple combinations' potentials is sufficient for guiding the search direction. To estimate a combination's potential, there is no need to train a model on the whole dataset to completion with full precision. Instead, we can process a sample of the dataset, conduct lower-precision computations, and perform fewer iterations of going through the training set without waiting for all parameter values of the model to fully converge.

To this end, and to expedite the search process, we perform progressive sampling, filtering, and fine-tuning to quickly reduce the search space. We perform fast trials on a small sample of the dataset to eliminate unpromising combinations. We then expand the sample, test and fine-tune combinations, and progressively shrink the search space in several rounds (Figure 1). In the last round, we narrow down to a small number of combinations and choose the final combination from them.

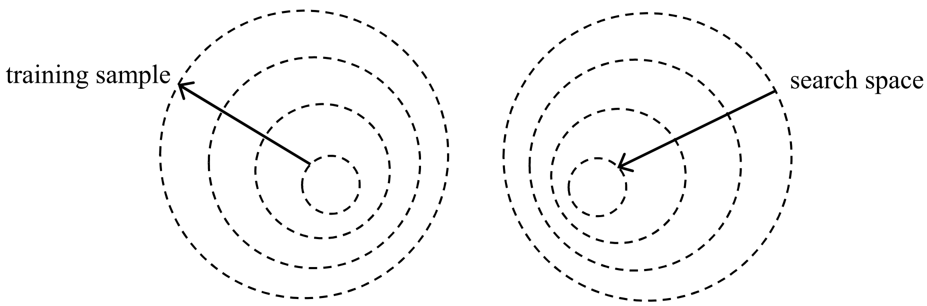


Figure 1. The relationship between the training sample and search space in our automatic model selection method.

How effective is our method? Compared to a state-of-the-art automatic model selection method on 27 benchmark datasets, on average our method cut search time by 28-fold and classification error rate by 11%. On each of these datasets, our method can finish the search process in 12 hours or less on a single computer. Not bad!

Next, let us move on to the second conundrum in machine learning: understanding models' prediction results.

People have raised many concerns regarding machine learning and its impact on society. A particularly serious concern involves most machine learning models' lack of transparency. A model can be inscrutable. If the means by which a model makes a prediction is concealed, how can the prediction result be trusted? One cannot simply look under the hood of a deep neural network model to see how the model operates. The model's decision-making process is hidden in an indecipherable network of interconnected layers, each with possibly thousands of nodes. In other words, the model is a black box.

Predictive modeling is a key component of solutions to many healthcare problems. Among all predictive modeling approaches, machine learning methods often achieve the highest prediction accuracy. But, as most machine learning models give no explanation for their prediction results, their deployment is hindered. For a model to be adopted in typical healthcare settings, interpretability of its prediction results is essential. Giving explanations can help identify root causes for bad outcomes and targeted preventive interventions.

This concern of interpretability is not limited to healthcare. The lack of transparency of machine learning models raises new challenges for myriad social issues, for example in ensuring non-discrimination, due process, as well as understandability in decision-making. FAT/ML—the Fairness, Accountability, and Transparency in Machine Learning organization—has pointed out that "... policymakers, regulators, and advocates have expressed fears about the potentially discriminatory impact of machine learning, with many calling for further technical research into the dangers of inadvertently encoding bias into automated decisions."

For all these reasons, the European Parliament has adopted a law for the European Union, which grants citizens the right to obtain an explanation for algorithmic decisions that significantly affect them. When this law becomes effective in April 2018, significant penalties for noncompliance could ensue for large Internet, credit card, and insurance companies that routinely use black-box machine learning models for purposes such as personalized recommendations, computational advertising, and performing credit and insurance risk assessments.

Automatically Explaining Machine Learning Models' Prediction Results

To overcome this difficulty of non-transparency, we recently developed a method to automatically explain the prediction results of any machine learning model *with no accuracy loss*. Our method uses a data mining technique to give explanations. We observe that prediction accuracy and giving explanations of prediction results are frequently conflicting objectives. Typically, a model achieving high accuracy is a complex black box, whereas a model that is easy to understand, such as a decision tree, achieves low accuracy. It is difficult to obtain a model that achieves high accuracy *and* is also easy to understand. Our key innovation is to separate explanation from prediction by using two models concurrently, each for a different purpose.

The first model makes predictions to maximize accuracy. The second uses class-based association rules mined from historical data to explain the first model's results. Everybody understands rules. For each data instance whose dependent variable is predicted by the first model to take an interesting value, the second model shows zero or more rules. Each rule gives a reason why the data instance's dependent variable is predicted to have that value. In many applications, the first model's end users do not need to understand the model's internal working. Rather, the end users only see the first model's predictions and only need to know the reasons for these predictions. The rules provided by the second model provide those explanations.

How does this work in practice? We demonstrated our method on predicting type 2 diabetes diagnoses in adults. An example rule is that in the past three years, if the patient had ≥5 diagnoses of hypertension AND prescriptions of statins AND ≥11 doctor visits, then the patient is likely to have a type 2 diabetes diagnosis in the next year. Our method explained the prediction results for 87% of patients whom the first model correctly predicted to have type 2 diabetes diagnosis in the next year.

Summary

We have surveyed here two of the foremost difficulties faced by machine learning model developers: selecting a machine learning algorithm and its hyperparameter values, and bringing transparency to the resulting model's prediction results. We present novel solutions to both challenges. Taken together, our methods bring high efficiency to machine learning model selection and provide automatically-generated explanations for any model's prediction results. These innovations make machine learning more accessible for critical application development, particularly in places where machine learning expertise is scarce, and help fulfill upcoming regulatory requirements.

To Learn More

See the two full text articles by Gang Luo *et al*. at http://www.researchprotocols.org/2017/8/e175/ and https://link.springer.com/article/10.1186/s13755-016-0015-4.

About the Authors

Gang Luo is an Associate Professor in the Department of Biomedical Informatics and Medical Education at the University of Washington, Seattle, WA, USA. He worked at IBM T.J. Watson Research Center and the University of Utah before. He received the BSc degree in computer science from Shanghai Jiaotong University, Shanghai, P.R. China and the PhD degree in computer science from the University of Wisconsin-Madison, Madison, WI, USA. To learn more about his work, visit his homepage at http://pages.cs.wisc.edu/~gangluo/.

John Schroeter is publisher at TechnicaCuriosa.com, the home of *Popular Electronics, Mechanix Illustrated,* and *Popular Astronomy* magazines. He also consults in the field of high-performance computing for deep learning applications.