Stats AI

The Impact of Bayesian Methods on AutoML

Hussain Abbas, MSc
CEO & Chief Data Scientist at Stats AI
PhD Student in Statistics & Data Science at University of Nevada, Reno

Stats AI

What is AutoML?

AutoML is the process of automating the end-to-end Machine Learning pipeline.

A Machine Learning pipeline consists of:

- Business problem definition
- Data gathering
- Data cleansing
- Data pre-processing
- Feature extraction
- Feature engineering
- Feature selection
- Algorithm selection
- Hyperparameter optimization

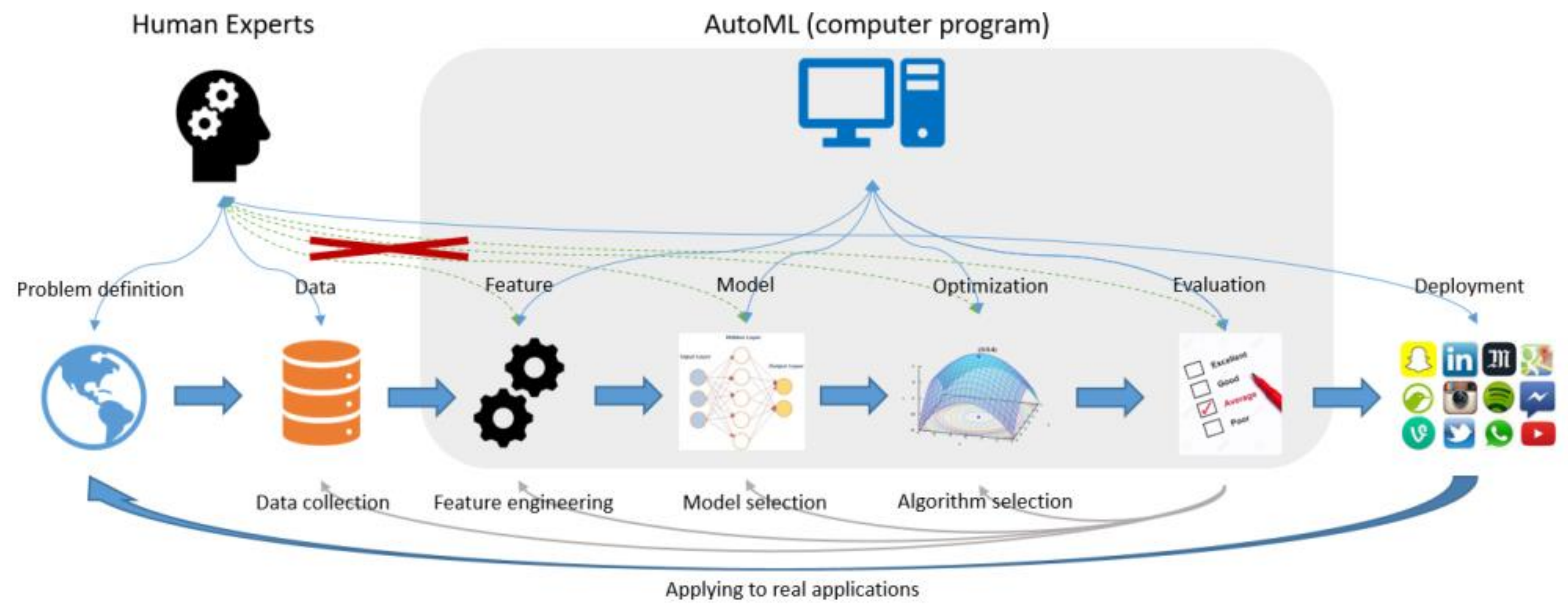

Fig. 1. To use machine learning techniques and obtain good performance, humans usually need to be involved in data collection, feature engineering, model and algorithm selection. This picture shows a typical pipeline of machine learning application, and how AutoML can get involved in the pipeline and minimize participation of humans.

A Taxonomy of Current Approaches to AutoML

Current approaches to configuration selection can be described as a structure consisting of an outer and an inner loop:

- The outer loop selects the candidate model and the inner loop searches through the set of hyperparameter choices for the model.
- As this process can be quite computationally burdensome, a method that can reduce this burden is preferred.
- Bayesian Optimization is such a method.

Open-Source AutoML Libraries

Current AutoML frameworks seek to automate a subset of the above tasks.

Some of the most popular frameworks are:

- Auto-WEKA (Java/GUI)
- Auto-SKlearn (Python)
- TPOT (Python)
- Auto-Keras (Python) Neural Networks only

What they use under the hood:

- Bayesian Optimization (BO): Auto-WEKA, Auto-Sklearn
- Neural Architecture Search: Auto-Keras
- Genetic Algorithms: TPOT

How Bayesian Methods Play A Critical Role in AutoML

- Bayesian Optimization (BO) is a derivative-free sequential sample-based optimization (SBO) procedure that can best be described as a fire and forget method analogous to how a homing missile zeroes in on its target.
  - BO solves black-box optimization problems by forming a probabilistic model of the loss function.
    - Example black box problem: Find the configuration (algo + param) which minimizes cross-validation loss.
  - BO is derivative free since it does not require the use of derivatives.
    - This differs from gradient descent. which utilizes gradient information to find a minimum of a loss function.
- Advantages of BO:
  - BO excels in finding optimal solutions in highly conditional spaces (such as algorithm/parameter space) which are computational burdensome to evaluate, such as the CASH problem which Auto-WEKA seeks to solve.
  - BO is particularly useful in situations in which evaluation takes a long time (minutes or hours to evaluate).
- Area of Active Research:
  - BOHB (2018) finds a solution 20x faster than traditional BO (2016) and 55x faster than Random Search (2011).

How AutoML is poised to upend the Data Science Industry

- Current State: A typical Data scientist earns a salary of $120,000 yet spends 80% of their time gathering and cleansing data and only 20% of their time conducting actual science.
- Future State: AutoML obviates the issue and changes the game by enabling data scientists to go deeper and broader than hitherto otherwise without increasing labor costs.
- Rather than spending time writing code to create models, the data scientist now simply leapfrogs off what the AutoML has built!
- With this newfound time, the data scientist is now free to:
  - Focus on the conclusions generated by the AutoML system and communicating those conclusions to business stakeholders, thereby enabling them to make better decisions.
  - Iterate faster through the pipeline of planned projects and deliver results.

$$A^*_{\lambda^*} \in \underset{A^{(j)}\in\mathcal{A},\lambda\in\Lambda^{(j)}}{\operatorname{argmin}} \frac{1}{k}\sum_{i=1}^{k} \mathcal{L}\left(A^{(j)}_{\lambda}, \mathcal{D}^{(i)}_{\text{train}}, \mathcal{D}^{(i)}_{\text{test}}\right) \Rightarrow$$

How Auto-WEKA uses BO under the Hood

- Auto WEKA uses BO under the hood to solve the CASH (Combined Algorithm Selection and Hyperparameter optimization) problem, since BO is capable of iteratively solving it.
- To kick off the algorithm, an initial **configuration** (An Algorithm and its corresponding Hyperparameter set) is selected using Meta-Learning:
  - Meta Learning maps configuration performance to historical meta-data (data about data) from past data.
  - Logic: Applying a configuration to datasets with similar meta-data ought to result in similar performance.
- The Algorithm stops when it has found the **configuration** which minimizes cross-validation loss.

Visit us at statsai.com
Email: hussain.abbas@statsai.com
Phone: 267-970-4873