JMLR: Workshop and Conference Proceedings 50:43–49, 2016 PAPIs 2015

The Past, Present, and Future of Machine Learning APIs

Atakan Cetinsoy CETINSOY@BIGML.COM
Francisco J. Martin MARTIN@BIGML.COM
José Antonio Ortega JAO@BIGML.COM
Poul Petersen PETERSEN@BIGML.COM
2851 NW 9th, Suite D,
Corvallis, OR 97330

Editor: Louis Dorard, Mark D. Reid and Francisco J. Martin

Abstract

In this paper, we start off by summarizing the key evolutionary turning points of machine learning APIs and conclude by laying out our vision for the future of this key enabling component that can power tomorrow's ubiquitous intelligent systems.

Keywords: Machine learning, predictive API

Having set out to make machine learning more consumable, programmable and scalable back in 2011, BigML has come to realize the central importance of REST APIs in tackling this big challenge. This has given us reason to believe that as machine learning continues its journey originating from the laboratory into the real-world, APIs will dominate its future footprint more than anything else.

1. The Evolution of Machine Learning APIs

It has been a long time since IBM's then CEO correctly predicted Professor Arthur Samuel's public demonstration of his checker playing self-learning program would increase IBM's stock by 15 points back in the 1950s. The primary actors of machine learning in that early phase were mainly academics and innovators at very large corporations that could afford large R&D departments. Starting in the 1980s machine learning split from the discipline of statistics and became a field of its own. International Machine Learning Society's first workshop was held in 1980 and it helped spur the formation of a community around the topic.

After the genesis, the areas of focus shifted towards inventing new algorithms and other theoretical research followed by the popular themes of parameter estimation and scalability. Around the turn of the century automation and composability became of interest, while some groundbreaking new algorithms such as automated representation were keeping the ball rolling on the earlier themes (see Figure 1). The attempts to improve existing algorithms by addressing their weaknesses led to the development of more complex techniques such as boosting and ensembles among which bagging and random forest are especially noteworthy.

More recently, the focus has shifted towards applicability and deployability of machine learning systems. Since the ultimate goal is to solve real-world problems, the industry needs to concentrate on what needs to be done to make modern machine learning algorithms ready for real-life challenges. One of the driving factors in this new direction is the data deluge fueled by increasingly powerful cloud storage and computing technologies, that are

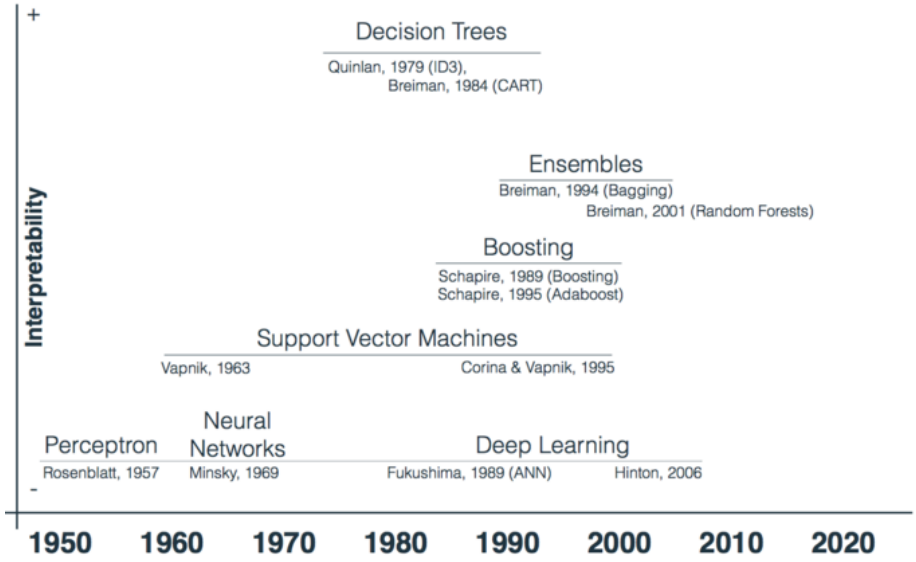

Figure 1: Evolution of Machine Learning Algorithms

able to cost-effectively process magnitudes higher amounts of data of ever greater variety, veracity, and velocity. Business media has promptly caught up on these innovations and the buzz generated in turn created heightened expectations for data-driven decision making powered by smarter enterprise applications. However, this promised land is remains to be fully surveyed for three primary reasons:

1. Real-world machine learning problems call for much more than picking the best algorithm,
2. Scaling machine learning to cope with the data explosion is a challenging process in itself, and
3. To complicate matters even further, existing set of machine learning tools were mainly designed for scientists — not developers.

2. Present Day Challenges

Let us cover these reasons in more depth. If we break down the typical machine learning project into its stages (see Figure 2), it becomes immediately obvious that the learning part is a single step out of many that all together make for a successful launch. High value predictive use cases must be identified with plenty of involvement from business analysts and domain experts. As a first step the chosen use cases must be stated as logical learnable problems. Data from disparate sources must be corralled, cleaned and conditioned before it gets fed to the tools that can efficiently train generalizable models.

As we turn our attention to the learning and the predicting stages of a typical machine learning project, we also need to keep in mind that most incoming data has inherent time-value that decays over time so it is crucial to act on it while it is still relevant. Rapid

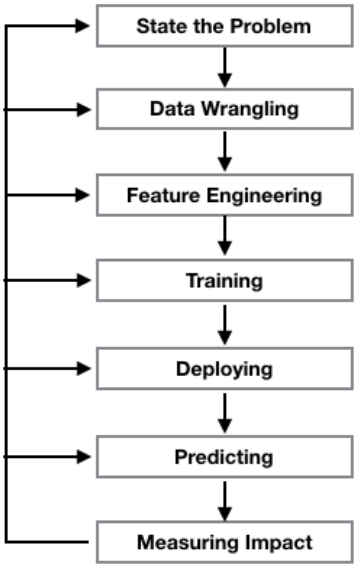

Figure 2: Typical Machine Learning Project Stages

turnaround of single stream of predictions is necessary but not sufficient for machine learning problems that call for a large number of parallel models. Scaling quickly becomes an equally big consideration. A good example is a collection of user specific fraud detection models that have to churn high volumes of accurate predictions with response times in microseconds in order to be deemed useful in a commercial setting. Successfully completing all the stages takes time and often requires multiple iterations before an acceptable predictive performance target is reached. This type of first hand experience leads machine learning experts to not only gain a new appreciation of the team effort required but it also further highlights that machine learning techniques are only as good as the impact it has on the real-world. This maxim shall serve as the guiding light of the machine learning community at large as we move forward. The third present day challenge we mentioned is the assertion that legacy machine learning tools have been built by scientists for scientists so they discriminate against developers that don't have a clue about all the different configuration parameters of a given algorithm. Even if one starts riding the parameter tuning learning curve, he is quickly faced with a 'Paradox of Choice' because of the exploding number of essentially similar algorithms. A recent paper (Fernandez-Delgado et al., 2009) compared a comprehensive list of 179 classification techniques arising from 17 different families against 121 data sets, which represent the whole UCI database (excluding the large-scale problems). The authors found that the random forest family of classifiers produced the best results overall even though different classifiers did best against different data sets. In a fast-paced business context, it is often more valuable to have a good solution fast instead of the perfect solution at an uncertain future date. Often, this makes it difficult to justify the time spent on optimizing many similar models due to diminishing returns. The fact that the Netflix Prize (Prize) winning ensemble methods never found their way into the company's real-

life production recommender system is yet another reminder of the need for feasibility in a business context defined by efficiency. Examples of currently utilized machine learning tools include both workstation-oriented pre-Hadoop software (Weka, R, Orange, KNIME, scikit-learn) and post-Hadoop ones (Mahout, Spark MLlib). However, the availability of larger datasets and more sophisticated data pipelines (e.g., AWS Data Pipeline) in the post-Hadoop era has done little to resolve the complexity issue, which still inhibits wholesale adoption. On the other hand, incumbent commercial data mining tools such as SAS and IBM SPSS suffer from the same problems except that they also happen to be much more expensive from a total cost of ownership perspective even as these vendors strive to fully adapt their portfolios to the cloud.

3. Future Scenarios

A promising approach in overcoming these sources of friction is exploring ways to further abstract machine learning by automating algorithm selection and parameter tuning while controlling any negative impact on eventual model performance. If we take a page from the historical arc of web application development, we observe that the evolution of REST APIs and JSON as a data interchange standard were ushered by industry leaders like Ebay, Salesforce and Amazon in the early 2000s. As these technologies reached maturity they increasingly defined the forefront of digital innovation in the commercial realm be it on desktops, mobile devices and for both business-to-consumer and business-to-business applications. Drawing parallels with the future evolution of machine learning tools, it is not too far-fetched to predict related APIs taking a lead role in defining their future trajectory. In fact, in the current decade we have already observed the launch of Google's Prediction API (2010) followed closely by BigML's inception in 2011. One might ask what is Hadoop's likely imprint on these new generation tools? Against the backdrop of massive tech media publicity taking up the wavelengths, it boils down to the provisioning of a more powerful on-ramp that supports more sophisticated data pipelines leading to more capable cloud-based machine learning applications. This means the big promises of distributed storage and distributed processing will remain unfulfilled unless the three speed bumps mentioned earlier are successfully addressed. Rather optimistically, we see this as a matter of time since a steady stream of new investment keeps flowing into Machine Learning as a Service (MLaaS) in the cloud most recently evidenced by the launches of Azure ML (2013) and AWS ML (2014). BigML's approach (see Figure 3) to its API turns each machine learning step to an end-point e.g. creation of a dataset, clustering, anomaly detection, model evaluation, batch predictions etc. Once provided with complete documentation developers feel right at home in accessing the REST API via the corresponding binding for their favorite programming language e.g. Node.js, Python, Java, Objective-C, Swift etc. On the other hand, our proprietary command line language BigMLer packages multiple machine learning tasks into a single command, which helps further abstract machine learning for non-specialists.

With concrete examples like that we can conclude that the confluence of the cloud and machine learning have taken on a new significance as cloud-based machine learning APIs can:

- Keep abstracting the complexity of machine learning algorithms,

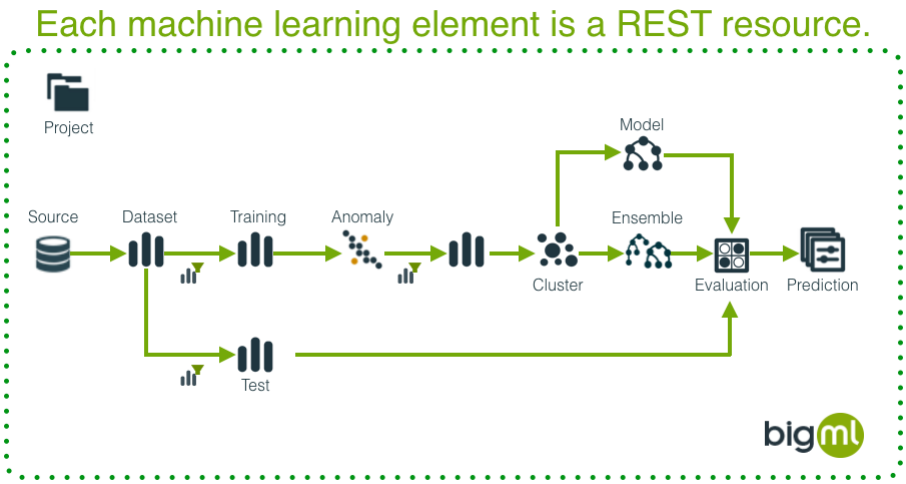

Figure 3: API-first Machine Learning

- Seamlessly manage the heavy infrastructure needed to learn from data and to make predictions at scale i.e. No additional servers to provision or manage,
- Easily close the gap between model training and scoring,
- Empower developers with full workflow automation,
- Add traceability and repeatability to all machine learning tasks for enterprises.

The combination of these enablers are paving the way to the democratization of machine learning. Without a doubt another democratization driver that cannot be overlooked is cost. Luckily, cloud-born machine learning tools with fully featured APIs come at a modest fraction of the total costs for equivalent tools from the previous era since those included many hidden fees and practically mandatory add-on modules to match the same level of capabilities. In the case of BigML, a computational bandwidth-based pricing scheme is on offer to specifically encourage rapid user adoption instead of a typical per user seat enterprise software pricing model.

Finally, freedom from any lock in effects and the preservation of future deployment options must be secured for full democratization to take place. We mean white-box models that can be exported out of the cloud environment they were created in so as to allow for the creators to port them over to a private cloud or any other run time environment of their choice. BigML has followed this flexible design to avoid potential customer backlash due to mounting MLaaS costs pertaining to large scale synchronous applications that require predictions to be served locally e.g. on a smartphone to minimize network latency and/or server-side workload.

We are hoping the arguments made so far have convinced the reader that the future of machine learning requires a mindshift from the obsession on algorithms to a more balanced viewpoint where the API is seen as "the product". Unfortunately, meaningful benchmark studies of current machine learning APIs are still few and far in between. This highlights the

heightened need for independent parties to test the existing set of machine learning APIs for us to accurately mark the industry's departure point circa 2015. Some of the dimensions that we recommend for consideration in such future studies are: number and type of algorithms supported, training speed, prediction speed, performance, ease-of-use, deployability, scalability and throughput, API design, documentation, user interface, SDKs, automation, time to productivity, importability, exportability, transparency, system dependencies, and price.

These design principles also summarize the internal architectural choices that BigML had to make in the last 5 years. Aside from internal architecture considerations a compendium of external and open REST APIs have surfaced (e.g. sentiment analysis, image processing etc.) pointing to a greater degree of specialization among machine learning APIs. This means an unprecedented new level of composability and many more interesting smarter applications made possible simply by mashing up these specialized APIs (see Figure 4). In the not so distant future, the need to start even a custom predictive application from scratch will be fully eradicated. A custom voice and visual assistant application for French hikers (Turner) was demonstrated at Gluecon 2014 as a case in point. This particular application lets hikers answer verbal and visual questions about plants and animals encountered on a hike e.g., "Est-ce toxique?" or "Is this poisonous?".

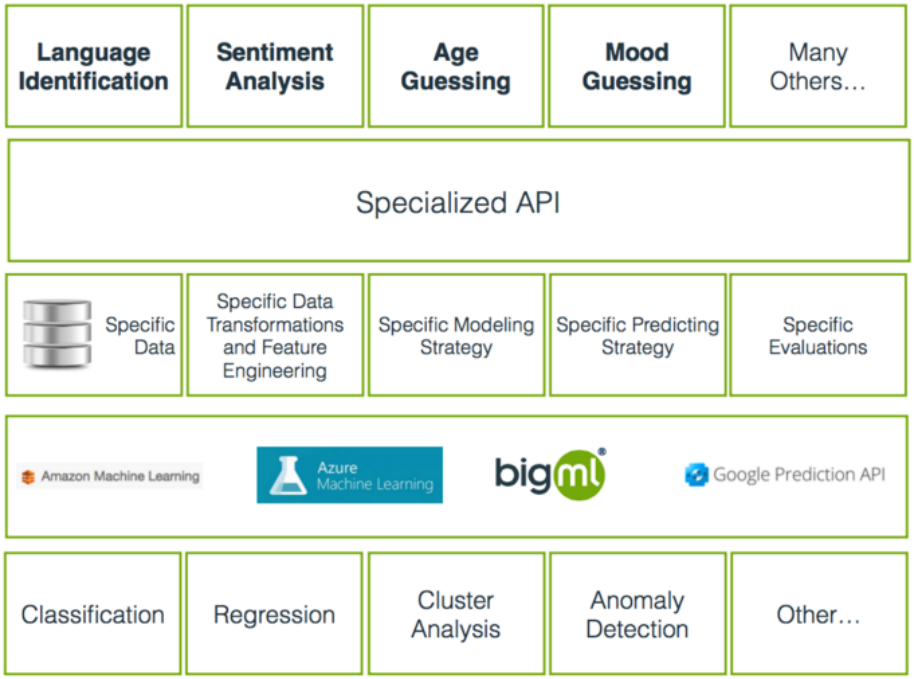

Figure 4: Machine Learning Turned into Specialized APIs

4. Conclusion

What is the end game here? In a sense the evolutionary path ahead may resemble what has happened with Web and mobile applications. They started off with custom scripts,

stove pipe integrations, screen real estate hogging UI controls and half baked million dollar launches. Over time, those disparate components left the center stage for standardized, pliable and robust development frameworks with built in separation of concerns, beautiful and easily customizable UI templates as well as agile and cost effective implementation methodologies. As cloud-based machine learning services mature by adding more and more features, they should stave off the trap of reintroducing unnecessary clutter and complexity that the previous generation of tools have been suffering from. Only then (and to the extent end-users will be given the opportunity to mainly be concerned with their predictive applications instead of the parameters and configurations of the underlying machine learning steps) will we reach full-on commoditization. Besides becoming an integral layer of the new cloud computing stack, machine learning will also transform data architectures such that implicitly predicted values will populate standard data model objects.
The onus is now on the technology providers to build machine learning APIs that are standardized, simple, specialized, composable, scalable yet cost effective in order to realize the goal of completely automated machine learning utilities, which will become an integral part of the modern software application development toolbox.

References

Manuel Fernandez-Delgado, Eva Cernadas, Senen Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 2009. URL http://jmlr.org/papers/volume15/delgado14a/delgado14a.pdf.

Netflix Prize. URL https://en.wikipedia.org/wiki/Netflix_Prize.

Elliot Turner. Enhancing your cloud applications with artificial intelligence. URL http://www.slideshare.net/alchemyapi/alchemy-api-enhancing-apps-with-aielliot-turnerjune-2014. GlueCon 2014.