Check for updates

Distributed collaborative prediction: Results of the machine learning contest

Matt Hall[1] and Brendon Hall[2]

Downloaded 09/02/17 to 186.228.165.244. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

The Geophysical Tutorial in the October issue of *The Leading Edge* was the first we've done on the topic of machine learning. Brendon Hall's article (Hall, 2016) showed readers how to take a small data set — wireline logs and geologic facies data from nine wells in the Hugoton natural gas and helium field of southwest Kansas (Dubois et al., 2007) — and predict the facies in two wells for which the facies data were not available. The article demonstrated with 25 lines of code how to explore the data set, then create, train and test a machine learning model for facies classification, and finally visualize the results. The workflow took a deliberately naive approach using a support vector machine model. It achieved a sort of baseline accuracy rate — a first-order prediction, if you will — of 0.42. That might sound low, but it's not untypical for a naive approach to this kind of problem. For comparison, random draws from the facies distribution score 0.16, which is therefore the true baseline.

We thought it would be fun to run a contest, asking readers: can you beat Brendon's result? We're pleased to report that lots of people had a go, and they did indeed beat it. After four months, the winner of the contest, LA Team, achieved a median accuracy of 0.6388, which represents a reduction in the error rate of 37.8% compared to Brendon's result. Their solution, along with all of the entries, is online at https://github.com/seg/2016-ml-contest.

What happened?

The contest received about 300 entries from 40 teams over four months (Figure 1). The teams were mostly individuals or pairs of scientists, and they came from all over the world, including the United States, Canada, Peru, the United Kingdom, Norway, Italy, Indonesia, and India. Especially exciting is that most of the participants were relatively new to machine learning, and many of them were even quite new to programming.

After the contest opened on 1 October, it took four weeks to get the first entry from CannedGeo, aka Bryan Page, a geologist in Denver. Bryan epitomized the kind of person we were hoping to reach with the contest: he was a relative newcomer to coding and to machine learning, and he embraced the "open" vibe fully, starting a conversation on GitHub about his strategy. Major kudos to Bryan for getting the ball rolling.

It was another two weeks before the eventual winners joined in — LA Team, consisting of Lukas Mosser (an Austrian petroleum engineer, now a PhD student at Imperial College London, UK) and Alfredo de la Fuente Briceño (a graduate petroleum engineer at Wolfram Research in Lima, Peru). LA Team were among the most prolific teams in the contest, along with Thanish (a programmer in Chennai, India). As a result, they achieved some of the lowest as well as the highest scores — they explored the solution space fully! Like Thanish, they also attempted several different algorithms, from random forests to deep neural networks. They were also the first team to use extreme gradient boosted trees, the model type that ended up dominating the top-ranked entries. Part of their solution is shown in Figure 2.

The exponential pattern continued: two weeks later, we had four teams, and there were 15 by late December. Fortunately for Matt, who was validating the entries — not only scoring them but running the code that generated them — the pace of new predictions did not get out of hand until the last few days in January. In the last 24 hours of the competition, six teams joined the contest, and we received more than 80 entries. The contest closed on 31 January at midnight, UTC.

The top-ranked teams at the end of the contest are shown in Table 1. The top three teams were awarded prizes, which included Raspberry Shake geophone kits, books from SEG Book Mart, machine-learning-flavored T-shirts, and other goodies. Congratulations to all of the winners, and kudos to all of the participants for having a go. Most of the entries explored novel aspects of the problem, and we were universally impressed with the creativity and enthusiasm of everyone involved. In turn, the phrase we heard over and over again was, "We had a lot of fun, and we learned a lot." Mission accomplished!

Lukas Mosser (left) and Alfredo de la Fuente Briceño (right) comprised the winning team in the machine learning contest introduced in *TLE*'s October 2016 issue.

Table 1. The top five teams at the end of the contest, with median accuracy scores from 100 realizations of their models. All of these entries implemented the XGBoost model in Python. You can see all of the teams, with deterministic scores, at github.com/seg/2016-ml-contest.

1	LA Team (Mosser & de la Fuente)	0.6388
2	PA Team (PetroAnalytix)	0.6250
3	ISPL (Bestagini, Tubaro & Lipari)	0.6231
4	esaTeam (Earth Science Analytics)	0.6225
5	SHandPR (Hall & Raghavan)	0.6200

[1]Agile.
[2]Enthought.

http://dx.doi.org/10.1190/tle36030267.1.

Downloaded 09/02/17 to 186.228.165.244. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

Models

There are many possible approaches to the task, which can be characterized as a small supervised, multiclass classification problem. In all, participants employed at least a dozen different types of model. Several of these showed up only once or twice — a couple of k-nearest neighbor models, a multilayer perceptron, an autoencoder, logistic regression — but a few model types were conspicuously more popular and/or successful:

- *Support vector machines* are fairly simple models that find the optimal *n*-dimensional hyperplane separating a subset of the data instances (the support vectors). This is the type of model that Brendon used in his tutorial, and many teams started with a support vector machine.
- *Ensembles of decision trees*, in particular *random forests* and *extreme gradient boosted trees*. These methods combine the predictions of large numbers of semirandom decision trees. They are often very effective in multiclass classification tasks, if you can avoid their tendency to overfit the training data. All of the top 10 models at the end of the contest were of this type.
- *Ensembles of models* implement a sort of democracy, combining models by either soft voting (the probability distributions are combined and the most likely facies wins) or by majority (the models vote and majority rules). In theory, these should

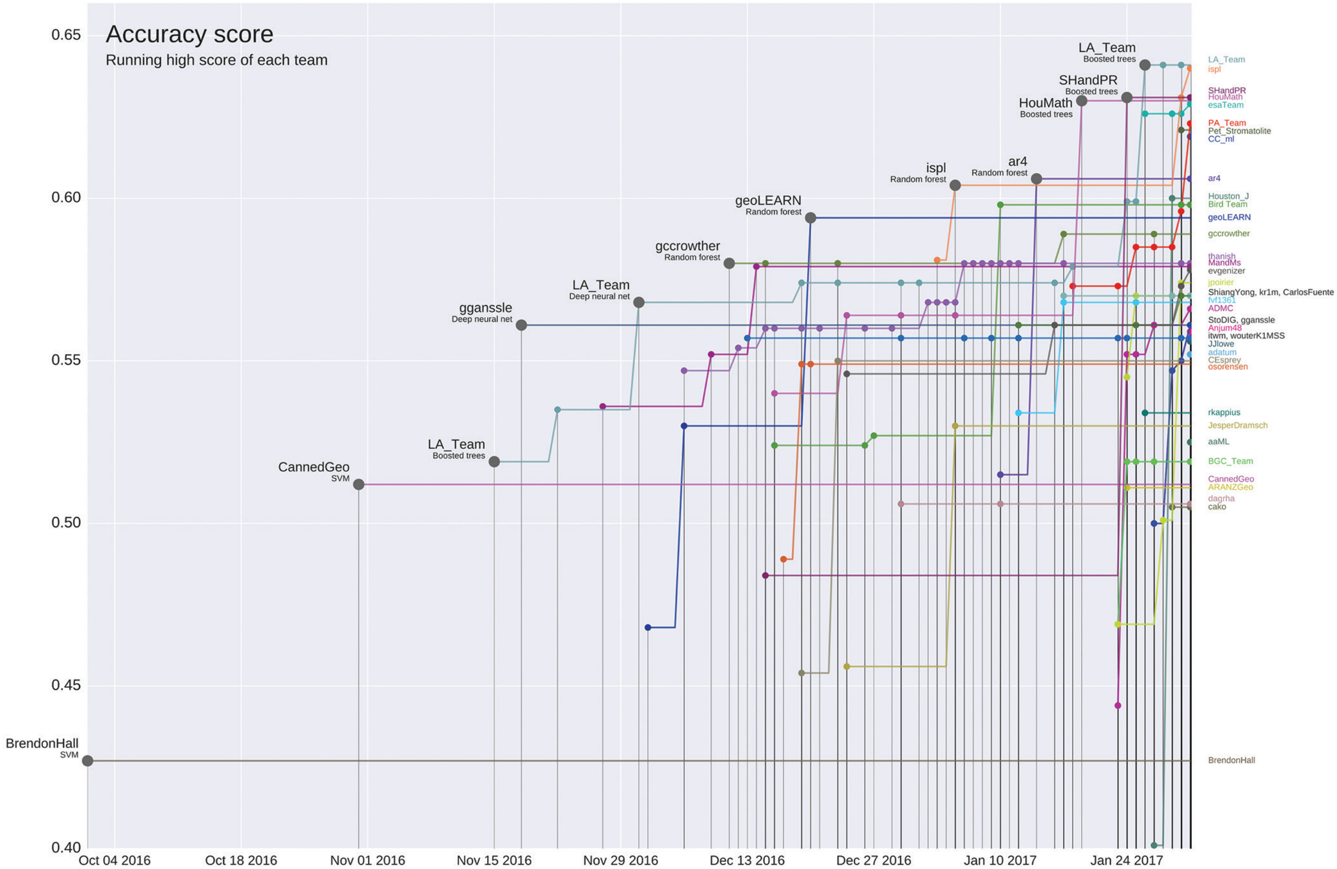

Figure 1. The progress over time of entries to the contest. Each team is represented by a colored line; the line shows the team's highest score up to that point. Each entry is represented by a vertical line and a dot. Changes in the first-place position are shown by a larger gray dot, annotated with the name of the team and the type of model they used. The scores shown are the deterministic scores. The final positions shown in Table 1 were determined stochastically.

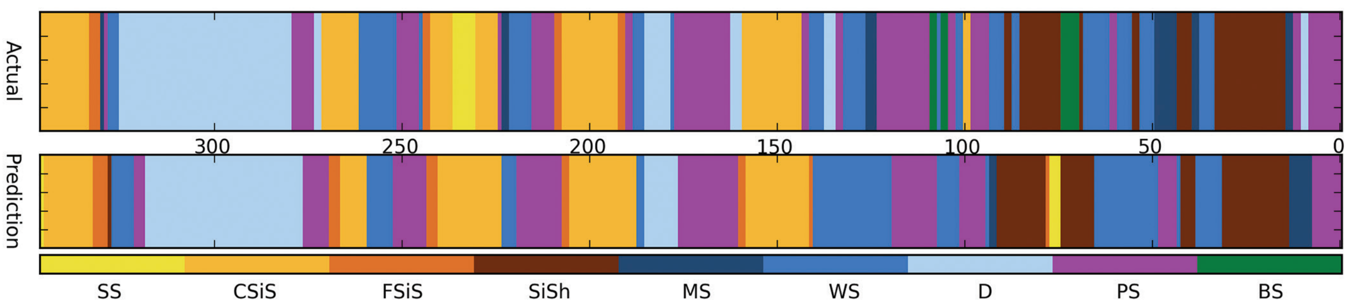

Figure 2. One realization of the solution for the Stuart well — one of the two blind wells. The true facies are shown on the left; the predicted facies are on the right. The facies labels are, from the left: nonmarine sandstone, nonmarine coarse siltstone, nonmarine fine siltstone, siltstone and shale, mudstone, wackestone, dolomite, packstone, and bafflestone.

Downloaded 09/02/17 to 186.228.165.244. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

outperform the models they contain, but in practice they can be unwieldy to set up.

- *Deep neural networks* are the hottest thing in machine learning today, popularized by their widely reported success in difficult tasks such as language translation and face recognition, and by the recent availability of open-source research software from companies like Google and Facebook. For various reasons, it was difficult to apply a deep neural network to this relatively small data set.

Beyond model choice, the most effective trick employed by the contestants was feature engineering. This involves using domain knowledge to create additional features that increase the accuracy of the classification model. The baseline model only used features from the nine wells in the original data set. The third-place ISPL team from the University of Milano (Paolo Bestagini and his teammates) devised a set of augmented features that were used by all of the top-ranked teams. These features exploit the fact that the data has some spatial correlation: the data are measurements of sediment columns, and the facies of a given interval is related to those above and below it. Thus features based on neighboring intervals were used to augment the feature set. This resulted in a significant boost in accuracy and provides welcome reassurance that human experts still have a valuable role.

Code

All of the code submitted by the participants is available at https://github.com/seg/2016-ml-contest. Indeed, one of the conditions of submitting an entry was releasing the code to reproduce the prediction. It's possible that this prevented some people from participating, but we have little doubt that the final result is better than it would have been if everyone's code had been secret.

It is difficult to estimate how much code people wrote for the contest. A very rough estimate of the time invested by the teams, based on the number of people contributing and an estimate (probably conservative) of 2 hours per person per submission, is 1760 hours, or about 1 year of full-time work.

All but five of the 40 teams programmed exclusively in Python, with four teams using the statistical programming language R, and one person (Graham Ganssle) taking the emerging Lua/Torch for a spin. The prevalence of Python could be due to its approachability, its extensive machine learning tools, or simply because the initial problem was framed in Python.

What did we learn?

In case you are thinking of running a contest like this, perhaps in-house or in a university course or in a technical society, we wanted to share some lessons we learned as organizers.

- *Withhold validation data from the participants.* We initially planned to hold one well from the training data blind, but one of the contestants pointed out that this would result in overfitting. We sought new blind data from Marty Dubois and Geoff Bohling at the Kansas Geological Survey, who graciously obliged.
- *Don't score entries against all of the blind data*, otherwise the overtraining problem reappears. Only validate against everything at the end.
- *Automate the validation process.* You must then be very clear about the required format.
- *Consider allowing stochastic realizations.* For example, allow teams to submit arrays containing multiple realizations of their prediction.
- *Provide an open, free, and accessible communication channel.* We used a combination of GitHub and Slack, but it wasn't perfect.
- *Engage actively with the participants.* Regular engagement with the teams helps keep the contest moving and is more welcoming for beginners.

Summary

This contest represented a distributed, collaborative technology development effort. Over the course of four months, teams from all over the world spent many hours refining their models. Tools such as GitHub and Slack allowed these teams to communicate, ask questions, and share ideas. The teams were able to iterate on their solutions and incorporate the ideas of other participants. This resulted in a robust and accurate facies classification algorithm, along with a well-documented exploration of the entire solution space, available to others to learn from and develop. This crowd-sourced approach to technology development is well proven (e.g., see kaggle.com) in a wide variety of industries and scientific disciplines. The proprietary nature of geoscience data in general may make this approach seem unrealistic for widespread adoption in the oil and gas industry. But open data sets are becoming more available, and this contest provides an example of the enthusiasm and talent that can be leveraged in this type of environment. There will be more! TLE

Acknowledgments

Thank you to Marty Dubois and Geoff Bohling of the Kansas Geological Survey for supporting the contest and providing the blind data. Thanks also to Evan Bianco for helpful comments on the manuscript.

Corresponding author: matt@agilescientific.com

References

Hall, B., 2016, Facies classification using machine learning: The Leading Edge, **35**, no. 10, 906–909, http://dx.doi.org/10.1190/tle35100906.1.

Dubois, M. K., G. C. Bohling, and S. Chakrabarti, 2007, Comparison of four approaches to a rock facies classification problem: Computers & Geosciences, **33**, no. 5, 599–617, http://dx.doi.org/10.1016/j.cageo.2006.08.011.

© The Author(s). Published by the Society of Exploration Geophysicists. All article content, except where otherwise noted (including republished material), is licensed under a Creative Commons Attribution 3.0 Unported License (CC BY-SA). See https://creativecommons.org/licenses/by-sa/3.0/ Distribution or reproduction of this work in whole or in part commercially or noncommercially requires full attribution of the original publication, including its digital object identifier (DOI). Derivatives of this work must carry the same license.