International Journal of Forecasting 36 (2020) 150–155

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Discussion

Machine learning in M4: What makes a good unstructured model?

Jocelyn Barker

Microsoft Corporation, United States

A B S T R A C T

The Makridakis Competitions seek to identify the most accurate forecasting methods for different types of predictions. The M4 competition was the first in which a model of the type commonly described as "machine learning" has outperformed the more traditional statistical approaches, winning the competition. However, many approaches that were self-labeled as "machine learning" failed to produce accurate results, which generated discussion about the respective benefits and drawbacks of "statistical" and "machine learning" approaches. Both terms have remained ill-defined in the context of forecasting. This paper introduces the terms "structured" and "unstructured" models to better define what is intended by the use of the terms "statistical" and "machine learning" in the context of forecasting based on the model's data generating process. The mechanisms that underlie specific challenges to unstructured modeling are examined in the context of forecasting, along with common solutions. Finally, the innovations in the winning model that allowed it to overcome these challenges and produce highly accurate results are highlighted.

1. Introduction

The M4 competition marked a turning point in the Makridakis competition series, as, for the first time, the winning model was one which would be classified colloquially as "machine learning". However, it was notable that the top performer was still tied to more traditional forecasting methods. The other applications of machine learning in the competition were largely unsuccessful. This is quite unlike the revolution that has been seen in image processing, where the top-performing methods are convolutional neural networks that take more or less raw data as inputs and are largely independent of earlier image processing technologies (Aloysius & Geetha, 2017).

Before discussing different types of models and their abilities to perform time series forecasting, it is important to define what these model types are. The terms "statistical" and "machine learning" models are used frequently, but do not seem to be defined clearly in the context of forecasting. Most "machine learning" models are maximum likelihood estimators, meaning that they are statistical in nature. Similarly, some traditional statistical models, such as autoregressive models, can be specified as a linear regression on lags of the series, and most people consider linear regression to be a machine learning model. Instead this paper will describe models as either "structured" or "unstructured", to better specify the underlying mechanism that seems to differentiate models labeled "statistical" from those labeled "machine learning". A structured model prescribes the data generating process (e.g. an autoregressive model only looks for additive relationships between lags of a series and its future values), while an unstructured model allows for relationships to be learned (e.g. a neural network creates its own features using non-linear combinations of its inputs). As such, a simple linear regression would be structured, but a decision tree would be unstructured.

It is not obvious where the winner of the M4 competition falls in this classification, as it contains elements that are seen frequently in both structured and unstructured modeling. At its highest level, the model emulates a Holt-Winters model with multiplicative seasonality, which is a structured framework. However, in place of the usual formula for the trend of the series, it substitutes a long short-term memory neural network model (LSTM), which is clearly an unstructured element.

Given this context, the question then becomes: "What was the nature of the winning model and what made it so successful compared to other models with unstructured

E-mail addresses: jocelyn.e.barker@gmail.com, jocelynbarker@verbsurgical.com.

https://doi.org/10.1016/j.ijforecast.2019.06.001
0169-2070/

elements?" At its core, the answer lies in two sets of challenges that unstructured models need to overcome: (1) the challenges in modeling the non-independent and identically distributed (iid) data that are inherent in the forecasting domain, and (2) the ability of the model to identify similar series in the dataset for efficient cross-series learning. This paper will explore the successful elements of the winning model and how they might function in other applications, as well as comparing it with other approaches to solving these problems.

2. A note on unstructured models in the M4 competition

For the most part, this paper will focus on the results from the winning entry and what can be learned from its success. Where applicable, methods outside the M4 competition will be discussed in addition to other methods within the competition. Particular note will be made where a method used an approach that clearly limited its potential success. However, it is difficult to make general comments about why the various unstructured models failed, for two reasons. First, the individual methods were generally very diverse, so it is difficult to compare where one method outperformed another and identify successful tactics, as the modeling methodologies are largely not comparable. Moreover, unlike many structured models, where excellent heuristics have been developed that allow even novices to create reasonable models with a few function calls (Hyndman & Khandakar, 2008), determining a good approach for unstructured modeling is frequently more experimental. The tools that one would generally use to diagnose problems in an individual unstructured model, such as training curves for neural networks, are not available, making it hard to say where the problem might have lain for an individual outcome.

Second, very few unstructured entries were submitted, so in many cases, extrapolating from the results would be drawing conclusions from a single instance. Even the benchmark "machine learning" models used in the M4 competition were not unstructured models: both used linear activation. Linear activation lies outside the constraints of the universal approximation theorem (Csáji, 2001), which forms the basis from which neural networks are able to approximate nearly any continuous function, and thus be unstructured. The ML benchmarks could have been specified equivalently as AR(3) models (see Appendix). The difference in accuracies between the two benchmarks is likely to be due to under-optimization, since the benchmark models have more complex optimization functions than a normal AR(3) model. Removing these unstructured entries leaves us with a total of four models that were self-labeled as "machine learning". Even comparisons outside the M4 competition are difficult, as these sorts of problems with linear activation and several other non-optimal unstructured modeling techniques, such as using local rather than global models, are common. For example, these problems were highly prevalent in a recent paper comparing "statistical" structured models and "machine learning" unstructured models (Makridakis, Spiliotis, & Assimakopoulos, 2018b), making its conclusions regarding the superiority of "statistical" structured models questionable. In this context, it seems more productive to focus on the successful advances seen in this competition and to limit our generalizations regarding the failures.

3. The winning model: structured or unstructured?

The confusion as to whether or not the winning model was entirely unstructured largely comes down to the role of data normalization. Data normalization is expected as a first step in most forms of unstructured modeling, with Z-score normalization or min–max normalization being nearly universal preprocessing steps. Normalization both improves the models' abilities to identify similar patterns on data of differing scales (Aksoy & Haralick, 2001) and improves the convergence of gradient descent algorithms (Ioffe & Szegedy, 2015). In time series data, it has been common to use more traditional time series techniques like seasonal and trend decomposition using loess (STL) as a preprocessing step and then model either components of the decomposition and residuals or residuals alone using unstructured modeling when evaluating unstructured modeling techniques (Makridakis et al., 2018b). For example, the ML benchmarks in the competition do the following: (1) preprocess the data by calculating the trend, seasonal, and residual components using a STL decomposition, (2) model the residual component of each series using a neural network architecture, and (3) re-trend and re-seasonalize by adding the trend and seasonal components back into the neural networks forecast. While there are other reasons why the "ML benchmark" models should not be considered unstructured models (discussed in Appendix), the models are not considered "hybrid" or "combined" approaches due to their use of these more traditional models as preprocessing steps.

The winning model effectively followed the same preprocessing method as the ML benchmarks, substituting the exponential smoothing class (ES) of models for the STL ones, but with one significant difference: rather than fitting the ES model once at the beginning as a preprocessing step, the seasonal and level components that normalize the time series data are fitted dynamically in the context of the LSTM, which accounts for the trend component of the ES model. The LSTM is effectively learning the preprocessing that is needed for each series along with its forecast values. This dynamic learning of the ES model components makes the final model less structured than if the normalization had happened independently of the model context. Since the ES components of the model act as data normalization and the LSTM component of the winning model is capable of learning not just the trend component, but also elements of the level and seasonality that are not captured by the ES model, the data generating process is learned, which makes the winning model unstructured.

When the creator of the winning model was asked to classify his model as either statistical or machine learning, he opted to describe it as a "hybrid" of the two approaches. This designation comes from the definition

that a statistical model is one that learns parameters locally (one series at a time), whereas a machine learning model learns the parameters globally (across multiple series). Since the winning model learns the seasonal and level components locally but the LSTM component globally, the winner considers the model to be a hybrid. This local/global distinction is fundamentally different from the structured/unstructured distinction drawn in this paper. The global/local distinction is drawn from the data that are used to train the model, meaning that the machine learning benchmarks used in the competition would be classified as statistical models, as a different model is trained for each series. On the other hand, the structured/unstructured distinction has to do with the way in which the data generating process is defined: is it defined a priori or learned from the data?

4. Modeling non-iid data

One of the key challenges of time series modeling using unstructured models is that, by nature, the data violate the underlying assumption in the models that the data are iid. If the data were independent, no traditional forecasting model that depended on lags of the series for forecasting would be successful. On occasion, series are stationary, meaning that they are identically distributed, but this is the exception, not the rule.

The structural elements of structured models are there to compensate for the known deviations from this iid assumption that most other statistical models, like machine learning models, make. Modeling these known deviations explicitly helps the structured models to extrapolate outside their training data. Unstructured models allow for the modeling of more flexible relationships between forecasts and their lags, making them generally optimized for interpolation. Since forecasting is necessarily an extrapolation problem, unstructured models may struggle unless the forecasted value can be framed into a more interpolation-friendly context.

4.1. Compensating for trend

One area in which the interpolation optimization may cause difficulty for unstructured models is modeling data with trend, where future values of the series may be larger or smaller than any value seen in the training data. This was an area where some of the entries in the M4 competition struggled. For example, one entry used a *k*-nearest neighbors (*k*NN) model to generate their predictions without using any (stated) preprocessing to detrend the series. Since a *k*NN model is a non-parametric model, it produces forecasts by averaging previous values from the training data. This means that the model is functionally incapable of creating forecasts with values larger or smaller than those it has observed previously, so any series with trend will have inaccurate results.

Preprocessing the series through detrending is one possible mechanism for overcoming this challenge. This will limit the accuracy of the results to the ability to extrapolate the trend to the forecasted values. An alternative mechanism is to forecast deviances (sometimes called differences or innovations) from the training data rather than predicting the value of the forecast, similarly to the "integrated" component of an ARIMA model (Bokde, Feijóo, & Kulat, 2018). This gives the forecasts a common baseline, creating a set of targets that resemble their training data more closely, making the problem more interpolation-friendly without the potential for reducing the signal through detrending. The winning model advanced this basic idea by modeling only the trend component of the ES model using an LSTM. In ES nomenclature, the trend component has a similar role to the deviance, representing the direction and magnitude of change in the series, while the level component shares more similarities with the trend component of an STL decomposition. The level component is the aspect of the model that is most likely to deviate strongly from the in-sample training data, whereas the trend should be more stable, since most series do not either increase or decrease in value exponentially with no damping. The LSTM can handle the more identically-distributed component of the data, while the level normalization supplies a mechanism for extrapolation.

4.2. Modeling seasonality

Similarly, seasonality causes deviations from the underlying model assumptions that need to be handled in order for good modeling to occur. As with the trend, attempting to remove the seasonality with deseasonalization is one possible option, with the same disadvantages. If this method is used, the disadvantages may be mitigated somewhat by including at least one seasonal lag in the model, to allow the model to compensate for seasonal components that are not accounted for in the deseasonalization.

A more common and generally robust method of accounting for seasonality in unstructured models is to provide an indicator variable for orienting the model within the seasonal cycle (e.g. month of year for monthly data; see Flunkert, Salinas, & Gasthaus, 2017). However, this is one area in which the design of the competition may have limited the success of the unstructured models. Unlike structured models, where a new model is generally created for each series, nearly all unstructured models benefit from cross-series learning, where patterns are learned across multiple series. Since the M4 competition did not provide dates for the series, it was not possible to align the series seasonally so that the same indicator represented the same point in the seasonal cycle for all series. In other words, while a local seasonal pattern can be learned on a series-by-series basis, global seasonal patterns such as end-of-fiscal-year effects for financial series are more difficult to align across series. Without alignment across series, trying to learn a global parameter for seasonality will add noise rather than signal. As a result, while many series probably showed similar seasonal patterns, at best these patterns were difficult for the unstructured models to learn. In fact, another recent forecasting competition cited these orienting seasonal variables as the most impactful element of the winning model (Kaggle, 2015).

While this limitation can be overcome, it will be unnecessary in nearly all applications, so very few mature models are likely to have a well-developed mechanism for compensating for seasonality without dates. Date-based seasonality modeling is not easy to remove, as it is frequently a model input that affects the architecture of the entire model. If a potential competitor had desired to submit such a model to M4, they would have had the following choices:

(1) Treat all of the series as non-seasonal data and submit an entry from their original model without seasonal modeling. This is likely to cause a drastic reduction in model accuracy.
(2) Design a new mechanism for modeling seasonality and incorporate it into their original model. This may require a greater time investment than is available in an industrial setting for a project that is unlikely to result in a product, and in addition, the new mechanism is likely to reduce the model accuracy, since it would be less well developed.
(3) Decide to sit out the M4 Competition.

Both options (1) and (2) carry the extra cost that if the model fails to perform well, it may be confused for the results of the mature model and damage the model's reputation. Based on the low M4 participation from tech companies that have invested in unstructured forecasting, it seems likely that many mature models chose option (3).

This limitation also highlights one of the strengths of the winning model in the context of the competition. Since the seasonal component of the model is fitted on a local, per series basis, there is no need for a global seasonal alignment of the series. It would be interesting to discover whether this local modeling of seasonality is still beneficial in a real-world application where dates would be available and the series would generally be more related to one another, or whether an unstructured global model of seasonality would improve the performance.

5. Modeling techniques and the "curse of dimensionality"

One culprit that is frequently blamed for unstructured models' inability to fit data properly is the so-called "curse of dimensionality", where the additional degrees of freedom that allow unstructured models to learn relationships in their data result in the models over-fitting the training data. The purported problem is that getting a dataset that is large enough to avoid over-fitting becomes intractable as the degrees of freedom increase, because the search space of the model increases exponentially with the degrees of freedom of the model, but the number of points to fit only increases linearly with the size of the dataset. However, this relies on the assumption that the data will be distributed throughout the entire search space of the model. In practice, it may not be possible for data to occupy the entire search space. The manifold hypothesis (Narayanan & Mitter, 2010) suggests that the data generating process lies in a lower-dimensional manifold within the larger feature space. In this case, the exponential increase in search space would not be seen.

It appears that the manifold hypothesis is likely to apply to time series applications. Data from the M3 competition were characterized by featurizing each series and plotting the first two principal components of those features (Kang, Hyndman, & Smith-Miles, 2017). A genetic algorithm was then used in an attempt to generate new synthetic series which would populate regions of the plot that the M3 competition did not cover. It was discovered that, while some new regions were able to be populated, there appeared to be clear boundaries that the algorithm could not cross, indicating that certain combinations of features may not be possible. Similar analyses were done of the M4 competition data, as well as of other competition datasets, and while these data were shown to populate the available regions more completely, they still were found to lie inside the boundaries observed in the original dataset (Spiliotis, Kouloumos, Assimakopoulos, & Makridakis, 2019; Spiliotis, Makridakis, & Assimakopoulos, 2018). t-SNE analysis on a variety of data sets, including the M-Competition datasets, also revealed densely-populated regions that were separated by unpopulated regions, all of which seemed bounded to a central region of the projected space (Kang, Hyndman, & Li, 2018). These observations of bounding in the engineered features are likely to apply to the learned features from an unstructured model as well.

5.1. Creating dense manifolds

The manifold hypothesis solution to the "curse of dimensionality" does highlight important features of successful unstructured modeling. For the solution to hold, the model first needs to be flexible enough to conform to the manifold in multidimensional space, meaning that high-dimensional structured models are unlikely to be successful. Second, the manifold where the data lie should be as low-dimensional as possible and as densely populated as possible. Three practices may be able to make this outcome more likely:

- Modeling many series with similar properties together is likely to generate a more densely populated, and thus better-defined, manifold (discussed further in Section 5.2). Those entries in the M4 competition that elected to use different unstructured models for each individual series may have had their effectiveness limited by the sparsity of the data within the search space.
- Data normalization is key to building an effective model, as normalized data are more likely to populate the same manifold space. Methods of normalization have been discussed in Section 4.
- Generating additional data by creating windows of training and testing data within each series (similar to rolling-origin cross-validation with a fixed window) may lead to a denser dataset, and therefore be preferable to creating a single train-test split for each series. This is an area in which the highest ranking self-described machine learning model could have been improved, as each series appeared in its training set only once.

It is important to note that all models, whether structured or unstructured, work within a manifold. In a structured model, the class of manifold is defined before the data are observed, and that class of manifold is then fitted to the training data. Limiting the manifold search space to a particular class enables the manifold to be defined by a smaller amount of data. The drawback is that if the data do not match that class of manifold, the model can never be fitted to it accurately. On the other hand, unstructured models learn the manifold by observing the data. This creates a more flexible model that can model more types of relationships, but needs more data to define the space. There is no manifold that can be modeled by structured models and cannot also be modeled by a sufficiently large unstructured model; however, if data are limiting, such as in local modeling scenarios, that manifold may be difficult for the unstructured model to find.

5.2. Grouping series for dense manifolds

The idea of generating a dense data manifold is somewhat at odds with the goals of the M Competitions. Since the goal of the M Competitions is to evaluate forecasting methods on a wide variety of data, diverse time series, which are less likely to occupy the same manifold space, are selected as targets for the competition. Compare this to most forecasting applications, where the forecasting target time series are likely to have some commonality in their origin, such as forecasting the demand for similar products at different store locations. The manifolds for forecasting applications are likely to be narrower and populated more densely than those for the competition data.

Some of the M4 entries attempted to group series with similar properties, which could potentially be used to identify data that were likely to form a well-defined manifold. For example, the second place entry used time series features to predict which model type was most likely to forecast each series accurately. For an unstructured approach, one might instead cluster series together based on these time series properties and train different models for each cluster of similar time series.

The winning model used an unstructured approach to create subsets of data that formed a coherent manifold, using the "ensemble of specialists" approach (Smyl, 2017). In this approach, N models are initialized and each series is assigned randomly to $n < N$ of those models for training. The models are trained for one epoch, then each series is evaluated on all N models and the models are ranked for each series based on their accuracy. The series are then reassigned to the n models with the highest accuracies for the next epoch of training. This process is repeated until the validation error begins to increase (early stopping). The result is that the series group themselves based on the models that can characterize them best, while the models tune themselves simultaneously to the series for which they are best suited. Between the innovative normalization using ES models and the ensemble of experts, the winning model managed to create well-defined manifolds for their models to fit.

6. Conclusions

Forecasting data violate the core assumptions of unstructured modeling methods, making it difficult to adapt these methods to forecasting problems. Compounding the problem, the qualitative nature of unstructured model optimization makes it difficult to tell the difference between a good model architecture that needs optimization and a bad model that is unlikely to yield accurate results. As the results of this competition have shown, researchers with a strong understanding of time series data and unstructured modeling practices can overcome these challenges to great effect.

Some have claimed that the results of this competition indicate that "pure machine learning" models are not competitive with structured models in forecasting (Makridakis, Spiliotis, & Assimakopoulos, 2018a), though no definition of a "pure machine learning" model is given beyond unguided self-labeling. Defining model classes more rigorously based on their underlying mechanisms means that the results of the competition instead highlight methods for generating successful unstructured models. The results of this competition seem to reflect those of the 2012 ImageNet Competition (Russakovsky, Deng, Su, Krause, Satheesh, Ma, et al., 2015), which many consider as the tipping point for deep learning in image processing. Deep learning architectures had been submitted in previous years and performed poorly, but in 2012 AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) not only won the competition, but showed an improvement over the runner-up of more than 10.8%. This competition saw some unstructured models struggle like those early image networks, but the winning unstructured model strongly outperformed the other models in the competition. It is the job of the forecasting community to adapt and develop the success of the winning model just as the imaging community did with AlexNet.

Appendix

To understand why linear activations used by the ML benchmarks are problematic, let us begin by defining the formula for the connection between any two layers of a neural network:

$$l_1 = W_1 x + b_1$$
$$a_1 = g_1(l_1)$$
$$l_2 = W_2 a_1 + b_2,$$

where the l_i values are the layers of the neural network, a_i is the activation for that layer, and x is either the input data or the activations from the previous layer. When we use a linear activation, we make $g_i(l_i) = l_i$, so we can simplify the above to:

$$l_1 = W_1 x + b_1$$
$$l_2 = W_2 l_1 + b_2.$$

Simplifying further, we get:

$$l_2 = W_2(W_1 x + b_1) + b_2$$
$$= W_2 W_1 x + W_2 b_1 + b_2.$$

However, since W_1, W_2, b_1 and b_2 are all just matrices of constants, we can rewrite $W_2W_1 = W'$ and $W_2b_1 + b_2 = b'$, so that the equation becomes:

$$l_2 = W'x + b'.$$

In other words, when activation is linear, the output of the second layer is just a different linear function of the first. This can be reapplied iteratively down all of the layers of the network, simplifying each layer to a linear combination of the layers before it. This does not change in a recurrent network, as the formula for a recurrent layer is:

$$a^{[t]} = g(W_x x^{[t]} + W_a a^{[t-1]} + b),$$

where $a^{[t]}$ is the activation of the layer at time step t, $x^{[t]}$ is the value of the target series at time t (i.e. the lagged value), W_x is the weight matrix for the lagged values, and W_a is the weight matrix for the previous time step's activation. Using linear activation starting from the second layer, we find:

$$\begin{aligned} a^{[2]} &= W_x x^{[2]} + W_a a^{[1]} + b \\ &= W_x x^{[2]} + W_a(W_x x^{[1]} + b) + b \\ &= W_x x^{[2]} + W_a W_x x^{[1]} + W_a b + b. \end{aligned}$$

Once again, the activation of the second layer is a linear combination of the lags of the series. If we continue to substitute iteratively into subsequent layers, by the end of the series we will find the formula for the final tth layer to be to be:

$$\begin{aligned} a^{[t]} = {} & W_x x^{[t]} + W_a W_x x^{[t-1]} + W_a^2 W_x x^{[t-2]} + \cdots \\ & + W_a^{t-1} W_x x^{[1]} + W_a^{(t-1)} b + W_a^{(t-2)} b + \cdots + W_a b + b. \end{aligned}$$

Once again, we see that this produces a linear combination of the lags in the series, meaning that, by the end of the time-steps, we will be able to define W' and b' values that can equally specify the network as $W'x + b'$. Since the "machine learning" benchmarks used three lags to fit the model, this means that they could have been specified equally as AR(3) models.

In practice, the model that results from a recurrent neural network that is trained in this way may not have the same results as an AR(3) model. This is because there is a single optimal parameter set in the AR(3) model that can be found using gradient descent along the convex surface of the model's optimization function. In the neural net example, though, there is no single optimal parameter set, as any combination of weight matrices that could be combined to create the same value of W' and b' would be optimal. The optimization function is non-convex and has multiple global minima, making it harder to find an optimal model. This under-optimization is likely to be the reason for the differences in accuracy between the two benchmarks.

References

Aksoy, S., & Haralick, R. M. (2001). Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, *22*(5), 563–582.

Aloysius, N., & Geetha, M. (2017). A review on deep convolutional neural networks. In *2017 international conference on communication and signal processing* (pp. 588–592). IEEE.

Bokde, N., Feijóo, A., & Kulat, K. (2018). Analysis of differencing and decomposition preprocessing methods for wind speed prediction. *Applied Soft Computing*, *71*, 926–938.

Csáji, B. C. (2001). Approximation with artificial neural networks (M.Sc. thesis), Hungary: Faculty of Sciences, Etvs Lornd University.

Flunkert, V., Salinas, D., & Gasthaus, J. (2017). Deepar: Probabilistic forecasting with autoregressive recurrent networks. ArXiv preprint. arXiv:1704.04110.

Hyndman, R., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software, Articles*, *27*(3), 1–22.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. ArXiv preprint. arXiv:1502.03167.

Kaggle (2015). Rossmann Store Sales, winner's interview: 1st place, Gert Jacobusse. URL http://blog.kaggle.com/2015/12/21/rossmann-store-sales-winners-interview-1st-place-gert/.

Kang, Y., Hyndman, R. J., & Li, F. (2018). Efficient generation of time series with diverse and controllable characteristics. *Tech. rep.*, Monash University, Department of Econometrics and Business Statistics.

Kang, Y., Hyndman, R. J., & Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, *33*(2), 345–358.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018a). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*(4), 802–808.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018b). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS One*, *13*(3), e0194889.

Narayanan, H., & Mitter, S. (2010). Sample complexity of testing the manifold hypothesis. In *Advances in neural information processing systems* (pp. 1786–1794).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Smyl, S. (2017). Ensemble of specialized neural networks for time series forecasting. In *37th international symposium on forecasting*.

Spiliotis, E., Kouloumos, A., Assimakopoulos, V., & Makridakis, S. (2019). Are forecasting competitions data representative of the reality? *International Journal of Forecasting*.

Spiliotis, E., Makridakis, S., & Assimakopoulos, V. (2018). The M4 Competition in progress. In: *38th international symposium on forecasting*.

Jocelyn's research focuses on the application of machine learning to problems in a variety of domains. In her Ph.D. in biophysics at Stanford, she used machine learning to identify subtypes of cancer. Now at Microsoft, she has spent the past two years developing quantitative methods for forecasting and forecast evaluation. Her pipelines forecast 100% of Microsoft's revenue every quarter and are consumed directly by the CFO, Amy Hood, who has described them as "an integral part of our financial planning and budgeting process".