Machine Learning Applied to Weather Forecasting

Mark Holmstrom, Dylan Liu, Christopher Vo
Stanford University
(Dated: December 15, 2016)

Weather forecasting has traditionally been done by physical models of the atmosphere, which are unstable to perturbations, and thus are inaccurate for large periods of time. Since machine learning techniques are more robust to perturbations, in this paper we explore their application to weather forecasting to potentially generate more accurate weather forecasts for large periods of time. The scope of this paper was restricted to forecasting the maximum temperature and the minimum temperature for seven days, given weather data for the past two days. A linear regression model and a variation on a functional regression model were used, with the latter able to capture trends in the weather. Both of our models were outperformed by professional weather forecasting services, although the discrepancy between our models and the professional ones diminished rapidly for forecasts of later days, and perhaps for even longer time scales our models could outperform professional ones. The linear regression model outperformed the functional regression model, suggesting that two days were too short for the latter to capture significant weather trends, and perhaps basing our forecasts on weather data for four or five days would allow the functional regression model to outperform the linear regression model.

INTRODUCTION

Weather forecasting is the task of predicting the state of the atmosphere at a future time and a specified location. Traditionally, this has been done through physical simulations in which the atmosphere is modeled as a fluid. The present state of the atmosphere is sampled, and the future state is computed by numerically solving the equations of fluid dynamics and thermodynamics. However, the system of ordinary differential equations that govern this physical model is unstable under perturbations, and uncertainties in the initial measurements of the atmospheric conditions and an incomplete understanding of complex atmospheric processes restrict the extent of accurate weather forecasting to a 10 day period, beyond which weather forecasts are significantly unreliable. Machine learning, on the contrary, is relatively robust to perturbations and doesn't require a complete understanding of the physical processes that govern the atmosphere. Therefore, machine learning may represent a viable alternative to physical models in weather forecasting.

Two machine learning algorithms were implemented: linear regression and a variation of functional regression. A corpus of historical weather data for Stanford, CA was obtained and used to train these algorithms. The input to these algorithms was the weather data of the past two days, which include the maximum temperature, minimum temperature, mean humidity, mean atmospheric pressure, and weather classification for each day. The output was then the maximum and minimum temperatures for each of the next seven days.

RELATED WORK

Related works included many different and interesting techniques to try to perform weather forecasts. While much of current forecasting technology involves simulations based on physics and differential equations, many new approaches from artificial intelligence used mainly machine learning techniques, mostly neural networks while some drew on probabilistic models such as Bayesian networks.

Out of the three papers on machine learning for weather prediction we examined, two of them used neural networks while one used support vector machines. Neural networks seem to be the popular machine learning model choice for weather forecasting because of the ability to capture the non-linear dependencies of past weather trends and future weather conditions, unlike the linear regression and functional regression models that we used. This provides the advantage of not assuming simple linear dependencies of all features over our models. Of the two neural network approaches, one [3] used a hybrid model that used neural networks to model the physics behind weather forecasting while the other [4] applied learning more directly to predicting weather conditions. Similarly, the approach using support vector machines [6] also applied the classifier directly for weather prediction but was more limited in scope than the neural network approaches.

Other approaches for weather forecasting included using Bayesian networks. One interesting model [2] used Bayesian networks to model and make weather predictions but used a machine learning algorithm to find the most optimal Bayesian networks and parameters which was quite computationally expensive because of the large amount of different dependencies but performed very well. Another approach [1] focused on a more spe-

Number	Name	Value
1	Classification	Clear
2	Maximum Temperature (F)	57
3	Minimum Temperature (F)	33
4	Mean Humidity	49
5	Mean Atmospheric Pressure (in)	30.13

TABLE I. Sample data from January 1, 2015, with the number, name, and value of each of the five features.

cific case of predicting severe weather for a specific geographical location which limited the need for fine tuning Bayesian network dependencies but was limited in scope.

DATASET AND FEATURES

The maximum temperature, minimum temperature, mean humidity, mean atmospheric pressure, and weather classification for each day in the years 2011-2015 for Stanford, CA were obtained from Weather Underground. [7] Originally, there were nine weather classifications: clear, scattered clouds, partly cloudy, mostly cloudy, fog, overcast, rain, thunderstorm, and snow. Since many of these classifications are similar and some are sparsely populated, these were reduced to four weather classifications by combining scattered clouds and partly cloudy into moderately cloudy; mostly cloudy, foggy, and overcast into very cloudy; and rain, thunderstorm, and snow into precipitation. The data from the first four years were used to train the algorithms, and the data from the last year was used as a test set. Sample data for January 1, 2015 are shown in table I.

METHODS

The first algorithm that was used was linear regression, which seeks to predict the high and low temperatures as a linear combination of the features. Since linear regression cannot be used with classification data, this algorithm did not use the weather classification of each day. As a result, only eight features were used: the maximum temperature, minimum temperature, mean humidity, and mean atmospheric pressure for each of the past two days. Therefore, for the i-th pair of consecutive days, $x^{(i)} \in \mathbb{R}^9$ is a nine-dimensional feature vector, where $x_0 = 1$ is defined as the intercept term. There are 14 quantities to be predicted for each pair of consecutive days: the high and low temperatures for each of the next seven days. Let $y^{(i)} \in \mathbb{R}^{14}$ denote the 14-dimensional vector that contains these quantities for the i-th pair of consecutive days. The prediction of $y^{(i)}$ given $x^{(i)}$ is $h_\theta(x^{(i)}) = \theta^T x$, where $\theta \in \mathbb{R}^{9\times 14}$. The cost function that linear regression seeks to minimize is

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} \|h_\theta(x^{(i)}) - y^{(i)}\|^2, . \tag{1}$$

where m is the number of training examples. Letting $X \in \mathbb{R}^{m\times 9}$ be defined such that $X_{ij} = x_j^{(i)}$ and $Y \in \mathbb{R}^{m\times 14}$ be defined such that $Y_{ij} = y_j^{(i)}$, the value of θ that minimizes the cost in equation 1 is

$$\theta = (X^T X)^{-1} X^T Y. \tag{2}$$

The second algorithm that was used was a variation of functional regression, which searches for historical weather patterns that are most similar to the current weather patterns, then predicts the weather based upon these historical patterns. Given a sequence of nine consecutive days, define its spectrum f as follows. Let $f(1), f(2) \in \mathbb{R}^5$ be the feature vectors for the first day and the second day, respectively. For i in the range 3 to 9, let $f(i) \in \mathbb{R}^2$ be a vector containing the maximum temperature and the minimum temperature for the i-th day in the sequence. Then define a metric on the space of spectra

$$\begin{aligned} d(f_1, f_2) = \sum_{j=1}^{2} \Big[& w_1 \mathbf{1}[f_1(j)_1 \neq f_2(j)_1] \\ & + \sum_{k=2}^{5} w_k \big(f_1(j)_k - f_2(j)_k\big)^2 \Big], \end{aligned} \tag{3}$$

where w is a weight vector that assigns weights to each feature. Since the first feature is the weather classification and the difference between classifications is meaningless, the squared difference has been replaced by an indicator function of whether the classifications are different. Define a kernel

$$\ker(t) = \max\{1 - t, 0\}, \tag{4}$$

and let $\text{neigh}_k(f)$ denote the k indices $i \in \{1, \ldots, m\}$ of the k spectra in the training set that are the closest to f with respect to the metric d. That is,

$$d(f^{(i)}, f) < d(f^{(j)}, f) \tag{5}$$

for all $i \in \text{neigh}_k(f)$ and $j \notin \text{neigh}_k(f)$, and $|\text{neigh}_k(f)| = k$. Furthermore, define

$$h = \max_{i\in\{1,\ldots,m\}} d(f^{(i)}, f). \tag{6}$$

Then, given the values $f(1), f(2)$ of the first two days of a spectrum f, the remainder of the spectrum $f(i)$ for i in the range 3 to 9 can be predicted as

$$\hat{f}(i) = \frac{\sum_{j\in\text{neigh}_k(f)} \ker(d(f^{(j)}, f)/h) f^{(j)}(i)}{\sum_{j\in\text{neigh}_k(f)} \ker(d(f^{(j)}, f)/h)}. \tag{7}$$

Training Set Year(s)	Test Set Year
2011	2012
2011-2012	2013
2011-2013	2014
2011-2014	2015

TABLE II. The four training sets and test sets used in the 4-fold forward chaining time-series cross validation.

The error of the estimator $\hat{f}$ is defined to be

$$\text{Error} = \sum_{i=3}^{9} \|\hat{f}(i) - f(i)\|^2. \tag{8}$$

A more useful error that will be used in lieu of this is the root mean square (rms) error, which is defined to be

$$\text{Error}_{rms} = \sqrt{\sum_{i=3}^{9} \frac{\|\hat{f}(i) - f(i)\|^2}{14}}, \tag{9}$$

and provides the standard deviation of the individual error terms.

EXPERIMENTAL

Since weather forecasting inherently involves time series, k-fold cross-validation is a poor technique to analyze whether our model will generalize to an independent test set. Instead, a 4-fold forward chaining time-series cross validation was performed, wherein the test set consisted of the data from the year immediately following the training set, as in table II. This method more accurately models the weather at prediction time, since the model is based on past data and predicts on future data. A learning curve can also be generated, providing a useful gauge of the dependence of the model on the training set size.

With this in mind, the parameters of the functional regression model were chosen to minimize the rms error in equation 9 averaged over all 4 test sets in table II. The weights $w_2 = w_3 = 1$ in equation 3 were chosen since we believed that deviations in the maximum temperature and the minimum temperature should carry equal weight. Since the functional form of the estimator $\hat{f}$ in equation 7 was too unwieldy to perform stochastic gradient descent on, an exhaustive grid search was instead performed to optimize the other weights w_1, w_4, w_5. Alternating exhaustive grid searches over the weights w and the number of neighbours k were performed to optimize each of these values. An initial exhaustive grid search was performed with large increments to obtain crude estimates of these weights, with the values of each weight taken from the range 0-50 in increments of 10. The number of neighbours was taken to be $k = 5$. This yielded initial estimates of $w_1 = 20$ and $w_4 = w_5 = 0$. $w_4 = 0$ was the optimum weight of the mean humidity presumably since humidity correlates poorly with the maximum temperature and the minimum temperature, and humidity would be a more useful determinant of precipitation. $w_5 = 0$ turned out to be the optimum weight of the mean atmospheric pressure since there were only small deviations in the atmospheric pressure which did not appear to be correlated with the maximum temperature and the minimum temperature. With this in mind, the mean humidity and the mean atmospheric pressure were removed as features.

The hyperparameter of the number of neighbours k was then chosen in a similar manner, with an exhaustive grid search over both constant values and values proportional to the data set size. Values of k in the range 5-50 in increments of 5 and values of k proportional to the data set size with proportionality constant in the range 0.05-0.50 in increments of 0.05 were considered. Taking k proportional to the data set size greatly outperformed taking k to be constant, and the optimum proportionality constant was 0.10. w_1 and k were then fine-tuned together with one final exhaustive grid search, taking w_1 from the range 15-25 in increments of 1, and the proportionality constant of k from the range 0.05-0.15 in increments of 0.05. This yielded a final value of $w_1 = 18$ and $k = 0.095|\mathcal{D}|$, where $|\mathcal{D}|$ is the number of data points.

RESULTS

The rms error for linear regression and the variation on functional regression are shown in table III. The rms error for a professional weather forecasting service is also included in the same table. Since data regarding the accuracy of professional weather forecasting services in Stanford, CA were not available, the data were instead taken from weather forecasts for Melbourne, VIC by the Australian Bureau of Meterology's Victorian Regional Forecasting Centre. [8] The learning curves for the linear regression and functional regression models are also shown in figures 1 and 2, respectively.

Day	Linear Regression	Functional Regression	Professional
1	5.039	5.252	2.612
2	5.157	5.734	3.244
3	5.300	5.914	3.618
4	5.379	6.068	3.708
5	5.446	6.221	4.522
6	5.566	6.211	4.883
7	5.642	6.329	5.062

TABLE III. The rms error in degrees Fahrenheit for the linear regression model, the functional regression model, and professional weather forecasting services.

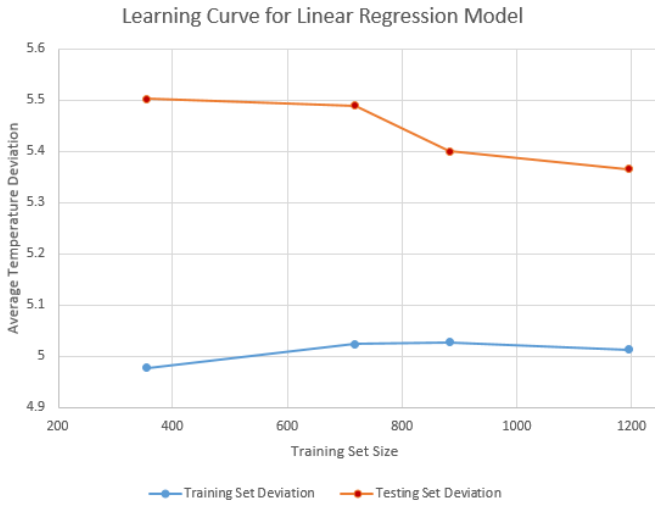

FIG. 1. The learning curve for the linear regression model, showing the rms error averaged across all seven days as a function of the training set size.

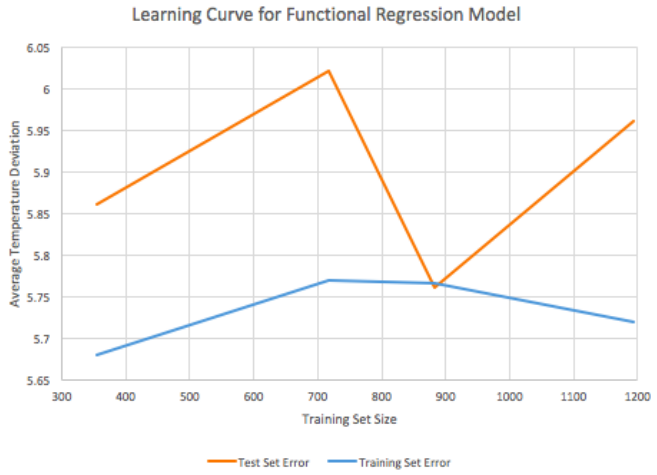

FIG. 2. The learning curve for the functional regression model, showing the rms error averaged across all seven days as a function of the training set size.

DISCUSSION

The professional weather forecasting service consistently outperformed our models across all seven days, with a large discrepancy in earlier days and a small discrepancy in later days. This was expected since the physical models of the atmosphere can be accurately solved for short time periods, but the instability of the physical models causes errors to accumulate quickly for longer time periods. On the contrary, machine learning algorithms are robust to perturbations in initial conditions, and over longer periods of time, perhaps our models would outperform the professional weather forecasting services.

The linear regression model also consistently outperformed the functional regression model across all seven days, with a small discrepancy in earlier days and a large discrepancy in later days. This is likely due to the fact that the forecasts were made solely based on the weather data for the past two days, which could be too short to capture significant trends in the weather. If this were true, then linear regression would be a better model than functional regression since there would be no trends to capture, and Occam's razor dictates that the simpler linear regression model is better. If the number of days on which the forecast is based were expanded to four or five days, then perhaps there would be evident trends in the weather that functional regression could capture, allowing functional regression to outperform linear regression.

As expected, linear regression proved to be a low bias, high variance model. The relatively low errors in the learning curve in figure 1 indicate that the model is an unbiased estimator. However, the large deviation between the training set error and the test set error that decreases slowly as the size of the training set increases indicates that linear regression is a high variance model. This is theoretically evident as linear regression is not robust to outliers, so collection of more data would improve the predictions of the linear regression model.

More interestingly, functional regression proved to be a high bias, low variance mode. The relatively large errors in the learning curve in figure 2 indicate that the model is a biased estimator. Again, this is likely due to two days being too short to capture any significant trends in the weather, so our model would be a poor predictor of future weather patterns. If the model were expanded to predict weather based on the past four or five days, perhaps this would suffice to capture trends in the weather, and the bias of the model would decrease. On the other hand, there is little deviation between the training set error and the test set error, with the test set error being even smaller than the training set error for a training set with three years' worth of data. This indicates that the model is low variance, so the model cannot be improved by collecting more data, only by changing our model to incorporate more days into each forecast.

CONCLUSION AND FUTURE WORK

Both linear regression and functional regression were outperformed by professional weather forecasting services, although the discrepancy in their performance decreased significantly for later days, indicating that over longer periods of time, our models may outperform professional ones. Linear regression proved to be a low bias, high variance model whereas functional regression proved to be a high bias, low variance model. Linear regression is inherently a high variance model as it is unstable to outliers, so one way to improve the linear regression model is by collection of more data. Functional regression, how-

ever, was high bias, indicating that the choice of model was poor, and that its predictions cannot be improved by further collection of data. This bias could be due to the design choice to forecast weather based upon the weather of the past two days, which may be too short to capture trends in weather that functional regression requires. If the forecast were instead based upon the weather of the past four or five days, the bias of the functional regression model could likely be reduced. However, this would require much more computation time along with retraining of the weight vector w, so this will be deferred to future work.

[1] Abramson, Bruce, et al. "Hailfinder: A Bayesian system for forecasting severe weather."International Journal of Forecasting12.1 (1996): 57-71.

[2] Cof?no, Antonio S., et al. "Bayesian networks for probabilistic weather prediction."15th Eureopean Conference on Artificial Intelligence (ECAI). 2002.

[3] Krasnopolsky, Vladimir M., and Michael S. Fox-Rabinovitz. "Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction."Neural Networks19.2 (2006): 122-134.

[4] Lai, Loi Lei, et al. "Intelligent weather forecast."Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on. Vol. 7. IEEE, 2004.

[5] Ng, Andrew. "CS229 Lecture Notes Supervised Learning" 2016.

[6] Radhika, Y., and M. Shashi. "Atmospheric temperature prediction using support vector machines."International Journal of Computer Theory and Engineering1.1 (2009): 55.

[7] "Stanford, CA" in Weather Underground, The Weather Company, 2016. [Online]. Available: https://www.wunderground.com/us/ca/palo-alto/zmw:94305.1.99999. Accessed: Nov 20, 2016.

[8] Stern, H. (2008), The accuracy of weather forecasts for Melbourne, Australia. Met. Apps, 15: 65?71. doi:10.1002/met.67