CS 466 Spring 2018 Final Project

Adam J. Stewart (adamjs5)

May 5, 2018

1 Introduction

The United States is the largest producer of corn worldwide. Since corn makes up a large percentage of our agricultural trade, accurate forecasts of crop yields are important for predicting economic growth, signing trade deals, and modeling insurance risks. The U.S. Department of Agriculture (USDA) releases historical data on crop yields, but not until the growing season afterwards. In-season predictions of crop yields are crucial for timely information relevant to these economic concerns.

Other researchers have attempted to predict crop yields in the past. The oldest models rely on modeling the growth process itself, trying to predict the yield based on knowledge of the crops themselves. Newer techniques include statistical models that ignore the underlying physiological behavior of plants and instead rely on training a model based on past data.

Yan Li, Kaiyu Guan, and Bin Peng developed a statistical model for corn yield prediction. This model includes the use of least squares fitting on lines, order 2 polynomials, and splines to predict crop yields [1]. Recent work done by Gro Intelligence, Inc. experiments with a machine learning approach to address the same task [2]. The purpose of this project was to compare various machine learning approaches to these statistical methods on the same dataset used by Li et al.

2 Data

A large dataset of environmental measurements as well as ground truth labels was used to train the machine learning model. Due to different sampling techniques, each measurement came at a different spatial and temporal resolution. Although I have access to daily climate data, the satellite data was limited to the frequency with which the satellite passes over the region of interest. The satellite data came at a fairly high spatial resolution, but the yield data was restricted to county-level resolution. In order to resolve this problem, all of the data was aggregated to the lowest common denominator: monthly intervals at county-level resolution. This is the same approach used by Li et al. and Gro Intelligence, Inc [1, 2].

Unpredictable cloud coverage resulted in many missing measurements. Since the model cannot handle NaNs in the data, all data points containing NaNs were removed from the dataset before training and testing. In particular, I only have access to satellite data for the 12 largest corn-producing states, so only these states are included in the training and testing datasets.

2.1 Climate Variables

In terms of climate data, I used temperature, vapor-pressure deficit (VPD), and precipitation to predict corn yields. All of the raw climate data was collected at a daily temporal resolution, but aggregated to monthly intervals. For each month, the model was trained on the following climate measurements:

1. Temperature
    - Maximum temperature
    - Minimum temperature
    - Average temperature
2. Vapor-pressure deficit (VPD)
    - Maximum VPD
    - Minimum VPD
    - Average VPD
3. Precipitation
    - Total precipitation

2.2 Satellite Variables

My satellite data came in two flavors: enhanced vegetation index (EVI) and land surface temperature (LST). Enhanced vegetation index can be thought of as a measure of the "greenness" of the satellite image, and serves as an indirect proxy for biomass growth. Maximum land surface temperature measured by satellites does not always agree with ground measurements, but is often at a higher spatial resolution, making it useful for training the algorithm.

2.3 Soil Variables

Environmental factors such as soil quality are also important in determining crop yield. In order to train the model, I used soil organic matter (SOM) and available water capacity (AWC) measurements at county-level resolution. Soil variables are assumed to be static, so there is only a single measurement for each county.

2.4 Ground truth labels

My ground truth labels came from yields published by the USDA after the harvest season ended. The original yield measurements are in bushels per acre (bsh/ac) but were converted to tonnes per hectare (t/ha) for comparison with previous results.

3 Methods

3.1 Machine Learning Model

I experimented with two common machine learning paradigms: linear regression and deep neural networks (DNNs). Although linear regression techniques have been used in the past for crop yield prediction, I wanted to reimplement this to provide a common baseline for comparison. Linear regression and DNNs were implemented using TensorFlow's built-in `LinearRegressor` and `DNNRegressor` estimators. [3].

3.2 Training

To train the model, I tried both stochastic gradient descent (SGD) with a batch size of 1 and mini-batch gradient descent with a batch size ranging from 16 to 128. I shuffled the dataset before selecting a random subset of the training data. I then repeated the same inputs for 10 to 100 epochs. Since neural networks are a non-convex optimization problem, SGD and mini-batch are useful techniques for finding global optima.

3.3 Cross Validation

In real world usage, the model will be trained on data from all previous years and used to predict the current year. To evaluate the performance of the model, I used two cross validation techniques: leave-one-out and forward. Leave-one-out cross validation allows the model to train on the entire dataset minus a particular year, then tests the performance on this particular year. Forward cross validation works more like the model will be used on real world data, where the model is trained on data from all years prior to the current year and tested on the current year. These cross validation techniques were used to evaluate the performance of the model for each year, and the final results were averaged over all years.

3.4 Metrics

Given a prediction for yields on the testing dataset, I needed a way to evaluate the accuracy of the model. I used three different performance metrics: root-mean-squared error (RMSE), Pearson correlation coefficient (r), and coefficient of determination (R^2). These values are defined as follows. Let y_i be a ground

truth label, and let $\hat{y}_i$ be a predicted value for the same data point x_i. If there are n data points in the testing dataset, then:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$
$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$
$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$
$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$
$$SS_{\text{res}} = \sum_i (y_i - \hat{y}_i)^2$$
$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

These metrics were gathered on a yearly basis, as well as for all years combined.

4 Results

Preliminary results show disappointing performance from both the linear regression and DNN models. Unfortunately, I ran out of time to tune all of the hyperparameters, so it is possible that with more time, I can achieve better results.

The best results for the linear regressor were achieved using a shuffle buffer size of 100,000, 50 training epochs, and a batch size of 16.

Table 1: Linear Regressor Prediction Accuracy

RMSE (t/ha)	**r**	**R^2**
1.840	0.619	0.380

The best results for the DNN regressor were achieved using a shuffle buffer size of 10,000, 20 training epochs, and a batch size of 1.

Table 2: DNN Regressor Prediction Accuracy

RMSE (t/ha)	**r**	$\mathbf{R^2}$
2.021	0.542	0.251

For comparison, Li et al. were able to achieve an RMSE of 1.027 (t/ha) and an R^2 of 0.823. See the Discussion section for plans for future work to improve these results.

5 Discussion

For the purposes of this project, I largely used the same dataset and techniques as Li et al. for a fair comparison between statistical models and machine learning models. However, there are several modifications that can be made to the model and new sources of data that can be added to improve prediction accuracy.

Currently, each county is treated as an independent data point, with no regional correlation. There are a couple ways that this can be improved upon. Obviously, crop yields are dependent on latitude and longitude, so one promising idea would be to translate the Federal Information Processing Standard (FIPS) codes for each county to the lat/long coordinates of that county. Another idea would be to build a convolutional neural network (CNN) that takes into account neighboring counties. If several neighboring counties were predicted to achieve higher than average crop yields, it would make sense for the county of interest to also achieve high yields, even if the data for this county contains significant noise that would otherwise lead to an erroneous prediction.

As mentioned in the Data section, each data source came at different spatial and temporal resolutions, and was aggregated to monthly county-level resolution. An interesting proposal would be to train off of the non-aggregated data. For example, if the first half of the month was incredibly dry, and the second half of the month was incredibly wet, one would expect very different yields than if the month was perfectly average, yet the total precipitation for the month would be the same. If it were possible to train off of every daily measurement at a higher spatial resolution, it should be possible to capture more of the features that influence crop growth. One way to handle this would be to train several neural nets, one for climate variables, one for satellite variables, and one for soil variables. Then, these separate models could be combined using an ensemble method like bagging or boosting.

Another major factor in the usefulness of the model is its robustness. As mentioned in the Data section, due to changing cloud cover and satellite trajectories, missing data is common in the dataset. Currently, the model cannot handle any NaN values, so these data points are removed from the dataset before training and testing. In order for the model to be used commercially, it needs to be made more robust in the face of missing data.

This summer, I will be working with Professors Jian Peng and Kaiyu Guan from the Computer Science (CS) and Natural Resources and Environmental Sci-

ences (NRES) departments to address these challenges. I hope to beat existing statistical models by exploring these ideas, including CNNs and non-aggregated data.

References

[1] Yan Li, Kaiyu Guan, and Bin Peng. Diagnostic analysis for improving yield prediction in statistical crop model (draft). 2018 (expected).

[2] Yiqing Cai, Kristen Moore, Aymn Elhaddad, Jerrod Lessel, Christianna Townsend, Hayley Solak, Nemo Semret, and Adam Pellegrini. Crop yield predictions - high resolution statistical model for intra-season forecasts applied to corn in the us (manuscript). 2018 (expected).

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.