**Statistics 578:** ***Statistical Learning in Data Science***

# Collaborative Yield Prediction with Recommender Systems

Benjamin Harlander
University of Illinois at Urbana-Champaign
Department of Statistics
bch3@illinois.edu

### Abstract

Due to variation in sample size, testing locations, and genetic properties, previous attempts to predict soybean yield relied on building a unique statistical model for each experimental variety. This approach can make large datasets of historic trials actually very small for modeling purposes. Alternatively, one can build a single statistical model for all varieties. While this provides a large sample size, this model may require a large number of degrees of freedom, violate assumptions, and fail to capture the unique environmental interactions of each variety. The research discussed in this paper explores the field of recommender systems in search of models that can capture the unique effects of a single variety while also utilizing observations from many varieties. Methods from the recommender systems domain are compared to traditional statistical learning methods including penalized regression and random forests.

## 1. Introduction and Motivation

In many retail applications, a company must choose a small subset of similar products to offer to the market from many potential candidates. This practice can be beneficial for both the customers and the retailer. The customer experiences a simpler decision making process at the time of purchase and the retailer gains efficiency in marketing, inventory management, and distribution. Choosing the right subset of products, however, can be a difficult task that involves many dimensions. The retailer must maximize profits while also satisfying a diverse customer base with variable preferences and appetites for risk.

This decision process is especially important for large agricultural seed retailers. Each season, the seed retailer selects a small number of seeds from hundreds of experimental varieties to offer to the market. The reputation of the retailer depends on choosing the varieties that will provide growers with maximal yield under uncertain weather conditions. The additional time required to have sufficient inventory when the growers are ready to plant only leads to further unpredictability for the season in question.

Like any other prescriptive analytics problem, a quality output is dependent on quality inputs. Accurate predictions of experimental variety yield under diverse weather and soil conditions are essential for creating a quality portfolio. Enlisting traditional statistical learning methods to model these varieties resulted in yield estimates with expected errors as high as 10 bushels per acre. With even the most complex models explaining just 10% of the total yield variation, it is unlikely that the retailer would be keen to use these predictions over a simple historic average for selecting a market portfolio.

The goal of this research was to test new methods for predicting yield that provide higher accuracy and more confidence when choosing an optimal soybean portfolio to offer the market. Specifically, this work was inspired by the methods supporting recommender systems. Recommender systems, or recommender engines, are described as models designed to "generate meaningful recommendations to a collection of users for items or products that might interest them" [4]. These recommendations typically involve predicting a user's preference as a binary or continuous rating and presenting the top-N items or products to the user.

Recommender systems have proven to be especially useful when the number of unique user-item interactions realized is substantially smaller than the potential combinations, leading to a sparse utility matrix. One example includes the Netflix Prize dataset where these algorithms provided exceptional performance with sparsity as high as 99%. The datasets used to model soybean yield also exhibit this sparsity, although to a lesser extent. Due to the high cost of yield trials, seed retailers are limited to testing experimental varieties in a small fraction of locations where the seeds could potentially be planted. This connection was another motivating factor for testing these models.

This paper will first provide a brief description of the datasets used for this research followed by an overview of the methods from the traditional statistical learning and recommender systems domains. Lastly, a series of numeric results will give an objective comparison of each method. Definitions of key terminology from agriculture and recommender systems are included in Appendix A.

## 2. Data Description

The data for this research was kindly provided by a research and development team at Syngenta AG, one of the world's largest agrochemical and seed retailers. As part of the 2017 Syngenta Crop Challenge in Analytics, teams were provided with historic yield performance in the "Experiment Dataset". This dataset contains over 82,000 yield trial observations from 175 experimental and commercial seed varieties tested in nearly 600 unique locations from 2009 to 2015 [1]. Each observation provides the variety name, measured yield, and the corresponding weather and soil measurements from the test site and year. Figure 1 provides a visual representation of the test site locations and frequency.

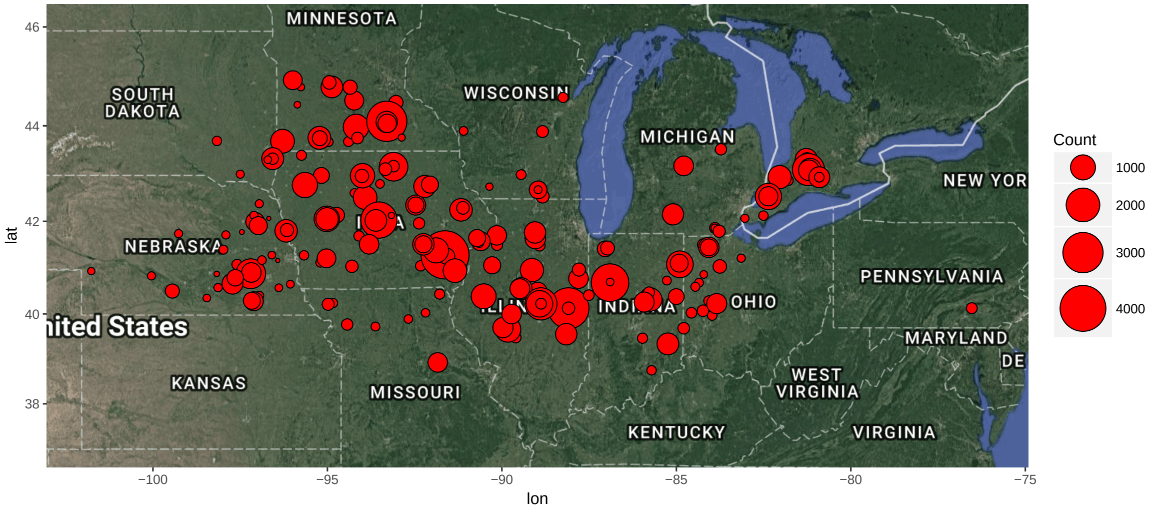

**Figure 1** **Test Site Locations**

## 2.1. Response

Two responses were extracted from this data for modeling purposes; one continuous and one binary. The continuous response is the observed yield from each trial yield measured in bushels per acre (bu/A). This represents the volume of harvested soybeans per unit of area. In addition to the observed yield from the experimental variety, each observation contains a Check Yield. Check Yield is the performance of elite commercial varieties planted at the same time and location as the experimental variety. This data point allows for a direct comparison at each test and for the creation of a binary response. If the experimental variety yield exceeds the Check Yield, it is considered a positive event (1), and the opposite is considered a negative event (0). For simplicity, this paper will refer to this binary response as BeatCheck. This alternative representation of the response allows one to model the overall competitiveness of the experimental variety against the current market leader. It also provides the opportunity to test recommender systems performance on a categorical response.

## 2.2. Features

The features, or predictive variables, in this dataset can be described by three categories: weather, soil, and experiment attributes. These categories are shown below with a more detailed description in Appendix A. A few additional features were generated from the set provided. With the exceptions of Soil Class and Year, all features were modeled as continuous predictors.

**Table 1 Features by Category**

| Weather | Soil | Experiment |
|---|---|---|
| Precipitation | Cation Exchange Capacity | Area of Soybean Coverage |
| Solar Radiation | pH | Days to Harvest |
| Temperature | Organic Matter | Productivity Index |
| | Clay, Silt, & Sand Mix | Year |
| | Soil Class | |

### 2.3. Sample Size by Variety

One important characteristic of this dataset was the variance in sample size and testing site locations by variety. Some varieties provided as few as 38 observations, while others provided nearly 4000. See Figure 2 for an illustration of this variation. This constrained not only which seeds could be modeled, but the types of models that are suitable for a given sample size. For this research, only the 55 experimental varieties with the largest number of samples were modeled.

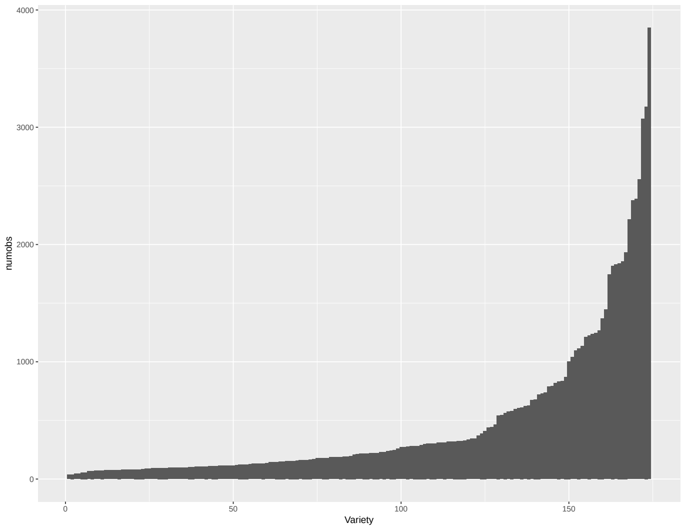

**Figure 2 Observation Count by Experimental Variety**

It is also important to note that each observation is not necessarily from a unique combination of covariates. For example, some varieties were replicated more than 100 times at the same time and location. With the resolution of the data provided, no difference in weather or soil conditions can be extracted. This is beneficial for calculating the irreducible portion of yield variance, but does not lead to a better predictive model. Replicated yield trials with identical covariates were aggregated by their mean. This same process was conducted for the BeatCheck response with the average of the 0-1 values rounded to the nearest integer. In the event of the average BeatCheck value being 0.50, the tie breaker was the historic Check Yield difference.

## 3. Methods

As mentioned in the introduction, the purpose of this research is to discover algorithms that outperform the traditional statistical learning methods. LASSO regression and Random Forests will represent the traditional methods. LASSO was chosen as a simple and interpretable model that provides a good baseline and Random Forests provides a more complex but flexible model that has proven to be a top performer in previous yield prediction tasks. Models selected from the recommender systems domain include user-based collaborative filtering, item-based collaborative filtering, and a few variants of matrix factorization.

**Table 2 Proposed Methods**

| Traditional Methods | Recommender Systems |
|---|---|
| LASSO Regression | User-based Collaboirative Filtering |
| Random Forests | Item-based Collaborative Filtering |
| | Matrix Factorization |

All methods are evaluated by their performance on a 30% holdout test dataset. To construct the test dataset, random stratified sampling was conducted by seed variety. A variety with 300 total observations would have 90 partitioned into the test, while a seed with 3000 total observations would have 900 partitioned into the test. This insured that all varieties were evaluated on equal terms. Recall that two unique responses were modeled for each method; the continuous observed yield and the binary BeatCheck. The continuous response model will be evaluated by the root mean squared error (RMSE) on the test set and the BeatCheck models will be evaluated using the area under the receiver operating characteristic curve (AUC). RMSE was chosen because it can be interpreted as the average number of bushels per acre the prediction deviated from the true value, and therefore easily evaluated by a seed retailer. AUC was chosen for the BeatCheck models because it is not sensitive to the class imbalance that many experimental varieties exhibit. The AUC also measures how well each model discriminates between positive and negative events. This is a useful property for recommender systems models that often rank order by the predicted probability and recommend the highest percentile items.

### 3.1. Traditional Methods

**3.1.1. LASSO Regression** LASSO, or least absolute shrinkage and selection operator, is a modified version of standard regression modeling. The name includes “shrinkage” and “selection” because LASSO adds a penalty term to the loss function that shrinks the size of the regression coefficients towards zero. The motivation behind LASSO is to create regression models with a higher prediction bias and lower prediction variance that leads to a lower overall mean squared error. LASSO regression can be applied to ordinary least squares regression and logistic regression. The updated LASSO loss function is shown for both regression types below.

(1) Continuous LASSO

$$\Sigma_{i=1}^{N}(y_i - \Sigma_j \beta_j x_i)^2 + \lambda \Sigma_j |\beta_j|$$

(2) Logistic LASSO

$$\Sigma_{i=1}^{N}((-log(1 + e^{\beta^T x_i}) + Y_i \beta^T x_i) + \lambda \cdot \Sigma_{j=1}^{p} |\beta_j|$$

The single added parameter, $\lambda$, determines the penalization magnitude. Cross validation using the training dataset is a simple way to find the optimal lambda that minimizes the mean squared error. 5-fold cross validation was used for this implementation. Once the optimal $\lambda$ is determined, the model is refit using all of the training data. This method was repeated for all 55 selected varieties and for both response types. Least squares LASSO (Equation 1) models were fitted for the continuous response and logistic LASSO (Equation 2) models were fitted for the binary BeatCheck response. This adaptive framework provides the means for each variety to determine the most important variables independently. Variable selection is important to avoid overfitting, especially with the small sample sizes available for this research. LASSO models were built with the glmnet package [3].

**3.1.2. Random Forests** Random Forests is a modeling method that produces many decision trees, where each tree is built on a bootstrap sample and a random subset of available variables is selected at each split. In theory, each individual tree has very little predictive power, but by averaging many individual trees, Random Forests creates a flexible and accurate model. This method requires the tuning of an important hyperparameter: m-try, or the number of variables randomly selected at each split. Similar to the $\lambda$ parameter in LASSO, 5-fold cross validation on the training dataset was enlisted to find the optimal m-try value. The optimal m-try can and does differ for each unique variety model. All Random Forests were built using the randomForest [4] package in R. This package provides methods for both continuous and binary response variables.

### 3.2. Recommender Systems

Recommender systems are often divided into two categories: content-based filtering and collaborative filtering. Content-based filtering involves creating a statistical user profile based on a history of purchases or item ratings. The ideas behind content-based filtering are similar to what was considered traditional methods for this research. A profile (statistical model) was created for each seed variety and this model was used to predict the rating (yield) for all locations. With these content-based methods explored, the recommender systems method this section will focus on include the set of algorithms from collaborative filtering.

Collaborative filtering can also be divided into two subcategories: neighborhood-based learning and matrix completion methods. To predict ratings on unseen content for a given active user, neighborhood methods find the the most similar neighbors and average the neighbor's ratings for the content the active user has not rated. This method can also be modified such that a neighborhood is created for each item instead of each user. The average ratings of users on similar items provide predictions for the active item. The next section will discuss how neighborhood-based collaborative filtering was applied to both location-based and variety-based models using the recommenderlab package [5].

Matrix completion methods differ from neighborhood methods in that all users and items are modeled simultaneously. The two matrix completion methods examined in this research include Singular Value Decomposition (SVD) and Factorization Machines (FM). Matrix completion methods are often very accurate, but are also computationally complex and require the entire model to be refit when a new user or item is introduced.

**3.2.1. Neighborhood-based Collaborative Filtering** In the context of recommender systems literature, entities which rate or interact with *items* are referred to as *users*. In the context of soybean yield prediction, each location rates or interacts with an experimental variety. This leads to the logical interpretation that *locations* map to *users* and *varieties* map to *items*. Therefore, these methods are the equivalent of traditional user-based and item-based collaborative filtering. Because soybean yield is highly variable from year to year, a location is actually represented by each unique location-year combination. This was a simple way to capture this temporal pattern.

Location-based collaborative filtering (LCBF) is presented first. For each location in the ratings matrix, similarity with all other locations was calculated. Pearson correlation and cosine similarity are two options explored for similarity metrics [5]. After similarity is calculated, the $k$ most similar locations were returned and considered the “neighborhood”. Predictions were then generated for all untested seed varieties by a weighted average of the yields of all $k$ users in the neighborhood. This same process is modified to create a variety-based collaborative filtering (VBCF) model. For each variety in the ratings matrix, similarity with all other varieties was calculated. Once again, the $k$ most similar locations were returned and averaged to create predictions for unseen locations. The weight in the averaging scheme for both models was determined by the similarity between the active location or variety and its respective neighbor. All models use a default value of 25 neighbors for averaging. Further details about this method are discussed in the recommenderlab vignette [5].

**3.2.2. Matrix Completion** SVD is a popular matrix completion method. For a ratings matrix $R_{m\times n}$, SVD decomposes this into an $m \times m$ matrix $U$, a $n \times n$ diagonal matrix $D$, and a $n \times n$ matrix $V$. SVD cannot handle missing values, so column imputation was used. The original ratings matrix could then be approximated by this decomposition while also creating predictions for all values in the test dataset.

(3)

$$\hat{R_{m\times n}} = U_{m\times m} D_{n\times n} V_{n\times n}^T$$

The key to avoiding overfitting lies in the diagonal matrix, $D$. The number of non-zero entries in $D$ dictates the rank of the approximation. For comparison, SVD was conducted with all non-zero values and then an optimal low-rank approximation was found with cross-validation.

Factorization Machines are a more general and flexible method for matrix completion. The libFM [6] package provides a simple framework for creating Factorization Machines. These methods work by estimating a bias term, $\omega$, for the global bias and each user and item while also estimating a set of latent factors, $v$, representing the interactions of each user-item combination. The equation below shows how a rating can be created from the sum of these parameters.

(4)

$$\hat{r}(x) = \hat{r}(u,i) = \omega_0 + \omega_u + \omega_i + \Sigma_{f=1}^{k} v_{u,f} v_{i,f}$$

Factorization machines also provide the opportunity to include contextual information. Up until this point, each location entity was actually modeled using the location-year combination. With factorization machines, the temporal pattern by year could be modeled explicitly. The equation below shows a new bias term, $\omega_t$, location-year, and variety-year interaction terms. This model is slightly more interpretable and achieved similar performance.

(5)

$$\hat{r}(x) = \hat{r}(u,i,t) = \omega_0 + \omega_u + \omega_i + \omega_t + \Sigma_{f=1}^{k} v_{u,f} v_{i,f} + \Sigma_{f=1}^{k} v_{u,f} v_{t,f} + \Sigma_{f=1}^{k} v_{i,f} v_{t,f}$$

Factorization machines in libFM have many methods for estimating each parameter including stochastic gradient descent, alternating least-squares, and Markov Chain Monte Carlo (MCMC).

MCMC was the default learning method and used for this research. MCMC provides the benefits that a learning rate does not need to be specified and regularization is not necessary.

## 4. Results

Numeric results are presented for the continuous response and binary BeatCheck response models separately. Each response type will first show the performance of the traditional methods followed by the recommender systems methods.

### 4.1. Continuous Response

**Table 3 Continuous Response Models**

| Category | Model | RMSE | Pseudo Rˆ2 |
|---|---|---|---|
| Traditional Methods | LASSO | 9.62 | 0.19 |
| | Random Forests | 9.64 | 0.19 |
| Recommender Systems | LBCF | 5.21 | 0.76 |
| | VBCF | 5.21 | 0.76 |
| | SVD | 5.04 | 0.78 |
| | FM | 4.83 | 0.8 |
| | FM with Time | 4.95 | 0.79 |

For the traditional methods, LASSO and Random Forests performed quite similarly. This is surprising as no non-linear transformations were fit for the LASSO models and Random Forests are afforded more flexibility with their decision tree basis. The recommender systems are able to predict yield with much higher accuracy. Many versions of LBCF and VBCF were tried by varying the similarity and normalization, but only the models with the best performance are shown. LBCF preferred normalized ratings and cosine similarity, while VBCF preferred non-normalized ratings and Pearson correlation as a similarity metric. SVD and Factorization Machines provide a further boost in performance with the basic FM achieving a pseudo $R^2$ as high as 0.80.

**Table 4 Binary Response Models**

| Category | Model | AUC |
|---|---|---|
| Traditional Methods | LASSO | 0.56 |
| | Random Forests | 0.58 |
| Recommender Systems | LBCF | 0.59 |
| | VBCF | 0.57 |
| | SVD | 0.609 |
| | FM | 0.66 |
| | FM with Time | 0.67 |

### 4.2. Binary Response

The binary response models show very similar results to the continuous models. One exception is that VBCF did not show as promising of results, with the Random Forests model showing a lower test AUC. Once again, LBCF preferred normalized ratings and cosine similarity, while VBCF preferred non-normalized ratings and Pearson correlation as a similarity metric. The best performers in this group are the Factorization Machine models with AUC values that are nearly 20% higher than the feature based methods.

## 5. Conclusions

Overall, the recommender systems methods show a significant boost in accuracy for predicting experimental variety performance. Factorization Machines showed the most promise for both continuous and binary yield models. 60% more yield variation was explained by the continuous FM models and cut the RMSE in half over the traditional methods. The higher AUC values of the binary FM models indicate a greater ability to discriminate between positive and negative yield outcomes.

This increase in predictive accuracy will provide seed retailers with a greater ability to recommend varieties and create optimal portfolios for the locations in which they have previously tested.

The downside of the recommender systems methods is that they are not able to extrapolate to new locations or varieties that have not been tested. This means that the traditional methods can still provide value even with a much lower predictive accuracy. A hybrid approach will create an optimal operational approach based on these results.

## 6. Appendix A

### 6.1. Terminology

Definitions of key terminology from agriculture and recommender systems is provided to assist the reader.

- **Yield:** Volume of soybeans collected at harvest measured in bushels per acre.
- **Check Yield:** Yield of elite commericial varieties planted at same time and location.
- **Variety:** Identifier for experimental seed with unique genetic properties.
- **Trial:** Individual soybean yield observation at specific time and location.
- **Location:** Experimental growing site maintained by seed retailer.
- **User:** Entity that interacts with items by making a purchase or declaring a rating as a measure of preference.
- **Item:** Static entity with which a user can interact. eCommerce products and movies are common examples.
- **Rating:** Implicit or explicit preference a user expresses towards an item.
- **Utility Matrix:** Matrix of user rows and item columns populated with ratings.

### 6.2. Feature Descriptions

- **Weather**

—Precipiation: Cumulative precipitation between April 1st and October 31st of the planting year.

—Solar Radiation: Cumulative solar radiation between April 1st and October 31st of the planting year.

—Temperature: Cumulative temperature between April 1st and October 31st of the planting year.

- **Soil**

  —pH: Soil acidity

  —CEC: Cation Exchange Capacity - Ability to transfer nutrients from soil to the plant.

  —Organic Matter: Level of organic nutrients necessary for growth.

  —Clay: Proportion of soil that is clay.

  —Silt: Proportion of soil that is clay.

  —Sand: Proportion of soil that is clay.

  —Soil Class: Class of soil based on sand, silt, clay composition.

- **Experiment**

  —Area of soybean coverage: Proportion of area dedicated to soybeans.

  —DTH: Days to harvest - Days between specified planting date and October 31st.

  —Productivity Index: Measure of a locations productivity and risk appetite.

  —Year: Planting year of experiment.

## References

[1] "Background." IdeaConnection. N.p., n.d. Web. 09 May 2017.

[2] Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. URL http://www.jstatsoft.org/v33/i01/.

[3] A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22.

[4] Sammut, Claude, and Geoffrey I. Webb. Encyclopedia of Machine Learning. New York, NY: Springer, 2011. Print.

[5] Hahsler, Michael. "Lab for Developing and Testing Recommender Algorithms [R Package Recommenderlab Version 0.2-2]." The Comprehensive R Archive Network. Comprehensive R Archive Network (CRAN), n.d. Web. 09 May 2017.

[6] Rendle, Steffen. "Factorization Machines with LibFM." ACM Transactions on Intelligent Systems and Technology 3.3 (2012): 1-22. Web.

[7] D. Kahle and H. Wickham (2013). ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. URL http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf

[8] H. Wickham (2009). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.