# A Matrix-Factorization Based Approach For Omnipotent-Yield-Predictor

Matrix-Factorization has been proved to be an effective approach for yield predictor. In Soybean Marathon Match 2, three out of the top five competitors adopted this approach. In this competition, I also used this method with some improvements. I’ll explain my solution in a incremental way in the later sections.

## **Method 1. Baseline Model**

**FILE : yield\_v1.cpp**

**SCORE :** **168238**

**MODEL:**

predict\_yield(v, l, y) = bias + bias(v) + bias(l, y) + p(v)\*q(l, y)

v : variety, l : location, y : year

**EXPLANATION:**

This is the standand matrix factorization method (see [1][2]).

**bias :** the average yield

**bias(variety) :** the average bias of this variety

**bias(loc,year) :** the average bias of this (loc, year)

**p(variety) :** the latent factors for variety

**q(loc, year) :** the latent factors for (loc, year)

Let’s see a simple example to better understand this model.

Say we know a variety V got a yield of 70 in location L in year Y.

How could we understand the value 70 ?

Now, say that 50 is the average yield of all variety.

Further more, V is bettern than an average variety, so it tends to get 10 bushels above the average.

On the other hand, (L,Y) is good land on which all variety could get 6 more bushels.

Finally, there may be some interactions better V and (L,Y). For example this variety likes more sunshine and this land has more sunshine, so the variety gets 4 more bushels on this land.

So, 70 = 50(average yield) + 10(variety) + 6(loc, year) + 4(interaction)

This method works pretty well, but to win, we need more.

## **Method 2.**

**FILE : yield\_v2.cpp**

**SCORE : 173534**

**MODEL:**

predict\_yield(v, l, y,e,r) = bias + bias(v,y)

+ bias(l, e, r) + bias(l, y) + bias(l, e)

+ p1(v,y)\*q1(l, e,r) + p2(v,y)\*q2(l,y) + p3(v,y)\*q3(l,e)

v : variety, l : location, y : year, e : experiment, r : rep

**EXPLANATION:**

In this method, we take experiment and rep into consideration.

But why this information is useful?

Let’s look at some data from DataTraining.csv :

9828,6210,1,307850,RR1,45.9,131,NULL,NULL

9828,6210,2,307850,RR1,62.7,125,NULL,NULL

9828,6210,3,307850,RR1,53.5,124,NULL,NULL

We could see that even the (v, l, y) is the same, the yield could vary drastically.

So take rep into consideration could help use better model the data.

For example, if we’re given the latter two lines of above data.

Method 1 would say :

the bias of (loc=6210, year=NULL) is high, so we should give a high estimation.

Method 2 woud say :

I couldn’t say the bias of (loc=6210, year=NULL) is high. Because this could be something caused by rep, or experiment. (for example, in this rep, we’ve better farms). I need to see more data so that I could decide the yield should be attributed to bias(l,y) or bias(l, e, r).

In this way, Method 2 could get better estimation than Method 1.

## **Method 3.**

**FILE : yield\_v3.cpp**

**SCORE : 174,277**

It’s essentially the same model as Method2. We just run Method2 five times and get the average.

Since we randomly initialize all the variables, if we run Method2 several times, we’ll get slightly different results.

By combining serval runs together, we could a get a slightly better result.

## **ABOUT THE CODE**

**FILE : yield\_v3.cpp**

MFMachine :

The class to perform matrix factorization. The code is mainly borrowed from [1]

OmnipotentYieldPredictor : BuildTrain :

Select a subset of training data, so we could run faster

OmnipotentYieldPredictor :SmoothExpYear

Since many training data have NULL for year. We guess the year from the data in the same experiment or the experiments with close id.

OmnipotentYieldPredictor : LearnMF

where we generate the data for matrix factorization and make our prediction.

The parameters (20, 0.5, 2e-2) are manully set according to local experiments.

20 : the dimension of the latent factor

0.5 : the regularizaiton term

2e-2: the learning rate

**Reference:**

( 1) a9108tc’s code for Soybean Marathon Match 2,

(2) http://www2.research.att.com/~volinsky/papers/ieeecomputer.pdf

(3) http://68.180.206.247/inner\_peace\_kddcup-camera\_ready.pdf