Multi-armed bandit for experimental plot selection

Yuji Saikai
Agricultural and Applied Economics
University of Wisconsin-Madison
427 Lorch Street
Madison, WI 53706-1503
Telephone: (608) 571-9556
E-Mail: saikai@wisc.edu

and

Paul D. Mitchell
Agricultural and Applied Economics
University of Wisconsin-Madison
427 Lorch Street
Madison, WI 53706-1503
Telephone: (608) 265-6514
E-Mail: pdmitchell@wisc.edu

Selected Paper
AAEA Annual Meeting
Washington, DC, August 5-7, 2018

1 Introduction

Big data has come to agriculture, attracting attention from the media, venture capitalists and academics, with applications of machine learning to farm management, especially crop production, receiving substantial interest (Coble et al. 2016). The large amount of data of various types now rapidly available create big data in agriculture and when combined with analytics and the capacity for precision agriculture, hold the potential for the next agricultural revolution. Conceptually, the process seems fairly straightforward – linking remote sensing and similar data with variable rate application of inputs and yield monitors to optimize the crop production system. However, discussions with farmers and others in the industry find that big data and data analytics have so far been a big disappointment, more promise and potential than practical applications that generate a positive return on investment for farmers (Colquhoun et al. 2017; Gandorfer and Meyer-Aurich 2017).

Analytically, the empirical goal is to identify the causal relationship between key inputs such as fertilizer, seed, or irrigation water and outputs such as crop yield, crop quality and/or profit. Given this production relationship, the system can be optimized and choices implemented using precision agriculture. Traditional approaches to identifying production relationships have included experiments with randomization and replication or combining observational data with structural or behavioral models (e.g., estimating supply and demand via two stage least square or using primal or dual approaches) or panel data methods (e.g., instrumental variables and fixed effects models). Typically, production scientists have relied on small plot experiments, while economists have used the latter two econometric approaches. Design, implementation and data collection for small plot experiments is relatively costly, typically only focusing on one or two inputs at a time, with results difficult to generalize to other seasons and locations. On the other hand, assembling the observational data and the required econometric analysis is beyond the capacity of the vast majority of farmers, and the analytical results are usually at such an aggregate level so as to not be relevant for (sub-)field level management decisions.

Machine learning is commonly mentioned as the solution for automating the collection and analysis of large data sets to quickly provide actionable recommendations to farmers (Coble et al. 2016). Machine learning is a broad term covering many types of algorithms, but at this time, their application to agricultural production contexts for decision making still remains rudimentary (Mishra et al. 2016; Shine et al. 2018; Sun et al. 2017). Athey (2017) highlighted the pervasive problem that machine learning algorithms have with identifying causal relationship from observational data – the type of data that most farmers generate or have available. However, farms can also purposely vary inputs from currently optimal levels as an experimental approach, but are typically not interested in experiments with full randomization and replication over a wide range of levels for key inputs. Wide adoption would require an automated experimentation and analysis process that did not imply high losses from substantial over use or under use of key inputs. Linking machine learning with precision agriculture offers a potential solution for lower cost and more widely used on-farm experiments to optimize crop management.

Farmers commonly experiment informally with new technologies, production techniques, crop varieties and other inputs, in essence trialing them on smaller portions of their farm to see if improvements result (Marra et al. 2003). The farmer's tradeoff is between earning income in the current season using standard production practices and technologies and learning the value of new or alternative production practices and technologies that will potentially improve income in future seasons. This conceptualization of the on-farm experimentation problem fits the classical tradeoff in reinforcement learning of exploitation versus exploration (Sutton and Barto 1998). Hence, we look to reinforcement learning for insights on algorithms to automate and to increase the efficiency of the on-farm experimentation process, as farmers would be more willing to adopt an easy-to-use and low cost process that generated a positive return on investment.

Multiple technical issues remain to be solved for this process and several interesting economic problems exist to address. The focus of this paper is developing and evaluating a machine learning algorithm to determine which parts of which fields to use for experimentation and which parts to use for typical crop production. We conceptualize the on-farm experimentation problem in terms of the multi-armed bandit problem from reinforcement learning, and then empirically evaluate a solution algorithm using existing small plot data. Our results demonstrate that plot selection for on-farm experiments guided by a multi-armed bandit machine learning algorithm can more efficiently estimate yield response curves than randomly placing plots in a field (the generally recommended method). We conclude by discussing some of the next steps needed for practical applications of our work.

2 Conceptual Framework

2.1 Classical multi-armed bandit problem

The multi-armed bandit is probably the simplest model to capture the exploration-exploitation dilemma (Berry and Fristedt 1985). A *K*-armed bandit problem imagines a slot machine to play with *K* arms. The player pulls one arm at a time and receives a reward. Over *T* number of plays, the player maximizes the sum total of expected rewards. As a slot machine, rewards are random, and so each arm defines a stochastic process $\{X_{i,t}\}$ for $i \in \{1, 2, \dots, K\}$, an independently and identically distributed sequence of random variables $X_{i,1}, X_{i,2}, \dots$ with an unknown mean μ_i. This situation requires a balance between exploring the arms to find those with higher expected rewards and exploiting the arm with the highest estimate $\hat{\mu}_i$ at any given period. A number of algorithms have been proposed and theoretically analyzed to solve this problem (Bubeck and Cesa-Bianchi 2012).

2.2 Plot selection as a multi-armed bandit problem

The following example adapts the multi-armed bandit problem as a model for a farmer choosing where to place experimental plots within a field for on-farm research. Suppose a farmer has precision agricultural equipment that allows control of inputs and measurement of yield, which together define a minimal management unit. For example, a farmer may have an 8-row planter with variable rate seeding and an 8-row combine that can measure yield along 40 feet of row, so that the management unit is 8 rows by 40 feet. Conceptually, the famer manages many fields and

each field is composed of many such management units and each of these units is potentially a small plot for conducting experiments for on-farm research. However, the farmer is primarily interested in earning income from growing a crop and experimentation is costly, both directly and in terms of opportunity costs from non-optimal input use or management.

Suppose a farmer wants to better estimate the soybean yield response to the seeding rate on the land he operates and for the varieties he plants. Thus he wants to choose *T* management units as small plots within a field of many such plots for an experiment in which the seeding rate is varied around the recommended level and the corresponding yield is measured. Each field is not homogenous, with soil characteristics such as soil type, organic matter, slope, depth to bedrock, and water-holding capacity varying across the field. The farmer must decide how to use this soil characteristic information to decide where to place each of these plots in his fields. We adapt the multi-armed bandit method proposed by Gutiérrez et al. (2017).

Suppose there are *M* soil characteristics or similar features for each field. Let $\boldsymbol{z}$ denote the *M*-tuple, each of whose coordinates takes a finite number of values $|\boldsymbol{z}(j)|$ for $j \in \{1, 2, \dots, M\}$. Then, there are in total $K = \sum_j |\boldsymbol{z}(j)|$ feature values, each of which defines an area in the fields. We conjecture that some areas of fields characterized by particular values of key soil characteristics or features provide better areas in which to place small plots for the purpose of estimating the yield response to seeding. For example, $\boldsymbol{z}(1)$ could categorize parts of the field by soil organic matter content, such as less than 3%, 3% to 3.5%, 3.5% to 4%, and more than 4%. In this case, $\boldsymbol{z}(1)$ partitions the field into four areas. Similarly, $\boldsymbol{z}(2)$ could categorize parts of the field by soil pH, such as less than 6.5, 6.5 to 6.7, 6.8 to 7.0, 7.1 to 7.3 and more than 7.3, partitioning the field into five areas. In this case, with these $\boldsymbol{z}(1)$ and $\boldsymbol{z}(2)$ features, we have 9 areas that likely overlap. We can regard each area of each field defined by these characteristics or features as an arm of the bandit. The farmer chooses an area (arm) from the *K* possible alternatives and sets a plot there, conducts the experiment and observes yield (sampling), and then determines whether the sample helped estimation of the yield response function (reward). Based on this logic, plot selection is formulated as a *K*-armed bandit problem.

For the empirical illustration here, we use the small plot data from Gaspar et al. (2015). These data are from replicated and randomized field trials conducted at nine locations throughout Wisconsin in 2012 and 2013. The experiments varied the seeding rate and measured yield, generating 1,148 observations in total. Soil pH was among the metadata collected for each location, with eleven unique values observed. Based on this information, we use *location* and *soil pH* as the variables defining the $\boldsymbol{z}$ features partitioning the field into areas. Thus, the farmer has *T* experimental plots to assign to a *location* and a *soil pH* from the following sets: *location = {Arlington, Chippewa Falls, Fond du Lac, Galesville, Hancock, Janesville, Lancaster, Marshfield, Seymour}*, and *soil pH = {6.0, 6.2, 6.3, 6.5, 6.6, 6.7, 6.8, 6.9, 7.0, 7.2, 7.5}*. These define the (overlapping) 20 segments of the training set below.

3 Solution Algorithm for the Multi-Armed Bandit Problem

Following Gaspar et al. (2015), we assume a negative-exponential function for the soybean yield response to the seeding rate: $y = \alpha\left(1 - e^{-\beta x}\right) + \varepsilon$. Here *y* is the observed yield, *x* is the seeding rate, α is the asymptotic yield maximum, β is the responsiveness parameter, and ε is a mean-zero error term. At each step $t \leq T$, our algorithm estimates the parameters α and β using the nonlinear least squares with all the sample observations collected up to that point $S = \{(x_1, y_1), \ldots, (x_t, y_t)\}$. The overall goal is to estimate the production function parameters α and β more accurately using a reinforcement learning algorithm to select where in the fields to place experimental plots, as opposed to creating a data set for estimation by placing plots randomly within fields.

Each iteration of the solution algorithm divides the 1,148 observations from the Gaspar et al. (2015) data into a training set of 574 observations (50% of the data), a validation set of 230 observation (20% of the data) and a testing set of 344 observations (30% of the data). Algorithm 1 summarizes one iteration of our solution algorithm as adapted from the algorithm proposed by Gutiérrez et al. (2017). Our algorithm takes the total number of plots *T* as an exogenous parameter, as well as the predefined field areas based on the features in $\boldsymbol{z}$ to partition the field into these areas (Line 1 of Algorithm 1). Variables and sets are initialized, including randomly drawing two observations from the training set as the initial sample (Line 2). For each step *t*, each field area is assigned a probability that locating an experimental plot in that area will give a sample point that will improve the estimation of the parameters α and β (Line 6). This probability is based on the number of times in previous steps *t* that placing a plot in that field area improved and worsened estimation based on the reward function (a_j, b_j, Lines 12-13). The field area with the largest probability of improving estimation is identified (Line 7) and an observation from that field area added to the sample (Lines 8-9). Estimates of the parameters α and β are then updated and the reward calculated (Lines 10-11).

Algorithm 1 incrementally selects *T* observations from the training set, while the standard randomization method for allocating experimental plots randomly selects *T* observations from the training set. The final result are two different sets of parameter estimates for α and β – one from the multi-armed bandit algorithm and one from the standard (randomized) method. As a measure of how well each algorithm performed in terms of estimating the yield response to the seeding density, the root mean squared error (RMSE) is calculated for both parameter sets using the testing set (not the training set). This process is repeated 10,000 times and the average RMSE calculated over these iterations. This process is completed for each value of *T*, with *T* varied from 5 to 70 with an increment of 5.

The success of our algorithm (and reinforcement learning in general) critically depends on defining the reward, as it signals the value of each arm and balances exploration and exploitation accordingly (Sutton and Barto 1998). The current application focuses on estimating the yield response curve and so uses the RMSE to determine the reward. Based on the use of the nonlinear least squares to estimate α and β, the binary reward for each step *t* is 1 if placing a plot in that field area and adding an observation to the estimation sample decreased the RMSE (success) and

-1 if it increased the RMSE (failure). Note that the RMSE for each step *t* is calculated using the validation set, not the training set used for the estimation or the testing set used for calculating the RMSE after the algorithm ends.

The process was conducted in Python, including the 10,000 iterations of our multi-armed bandit algorithm (Algorithm 1), the alternative algorithm using random selection of plots, and calculation of the average RMSE for both algorithms. Each iteration is randomized by a unique random seed, but the same set of 10,000 seeds is used for all *T*. The Appendix reports the full code.

4 Results

Figure 1 plots the average RMSE for both algorithms for estimating the soybean yield response to seeding rate as the number of observations (*T*) varies from 5 to 70. Not surprisingly, the relative performance of both algorithms is poor when *T* = 5, as they are both estimating 2 parameters with 5 observations. However, as the number of observations used increases, the average RMSE declines and gradually approaches the limit of about 20 bushels per acre. Our multi-armed bandit algorithm for selecting plots outperforms randomly selecting plots over the range of observations used. The performance difference between the algorithms is small at low values of *T* and again at high values. Furthermore, our algorithm stabilizes at an average RMSE of 20 at about *T* = 30, while the standard randomization method for plot selection does not stabilize until about *T* = 60 or even more. These advantages give some idea of the gains that using our algorithm could potentially achieve for improving the efficiency and lowering the cost of on-farm experimentation. Finally, as expected, the advantage of our algorithm shrinks as *T* becomes larger due to the diminishing advantage of the more targeted sampling method relative to the brute force approach of randomly adding more observations.

5 Discussion and Conclusion

This simulation analysis based on reusing existing data from randomized field trials demonstrates the potential for our algorithm to more efficiently allocate placement of plots for on-farm experiments than simple randomization. These promising initial results suggest that additional research is warranted, but many improvements are needed before practical application. We review some of these and potential approaches to address them.

First, our algorithm requires calculating a reward and therefore needs a separate validation data set. This simulation began with 1,148 observations from existing yield trials and separated them into a training set and a validation set. However, most farmers will usually begin with little or no data. In this situation, cross-validation potentially offers more efficient use of data (James et al. 2013). Specifically, suppose 20% of the samples available at any step are used for validation. Then, at each step $t \leq T$, the samples are randomly split into 80% for training and 20% for validation, the algorithm is run and recommends a field area for placement of a plot. This process is iterated many times, with each iteration recommending a field area for placement of the next

plot. The final recommendation for the next step is then the most frequently recommended field area over all iterations.

Second, our algorithm assumes a single plot is added at each step *t*, as it was originally developed for situations with rapid availability of new data from which to sample. Realistically, however, given the annual timeframe of crop production, farmers will want to add multiple samples at each step. If *N* plots are sampled for each step *t*, the algorithm could sample all *N* from the recommended field area identified, but more promising is to use the ordering of field areas based on the $\{\hat{q}_i\}$ (Line 6 of Algorithm 1) to allocate the *N* samples to multiple areas. Several possibilities seem promising, such as choosing all areas above a threshold value for $\hat{q}_i$ or using the $\{\hat{q}_i\}$ as weights to allocate the *N* samples randomly across all areas.

Third, our algorithm recommends where in a field to place plots to sample from, but it is silent about which seeding rates to use for the chosen plots. Given the objective of efficient estimation of the yield response curve, a natural extension is to develop an algorithm that also recommends the seeding rate to use for the new plots to sample from. The Gaspar et al. (2015) data used for this demonstration contains only six different values for the seeding rate (*x*), ranging from 40,000 to 140,000 seeds per acre. Thus, despite having a total sample size of 1,148, we fit the smooth curve $y = \alpha\left(1 - e^{-\beta x}\right)$ with very little variation in *x*. Just as it was possible to improve the efficiency of estimation using soil metadata and our algorithm to target placement of plots within fields, it seems likely that additional efficiency can be gained by selecting input levels more carefully. In general, the selection of experimental levels for continuous treatments has received little attention in the production sciences.

Fourth, recall that our algorithm assumes that some areas of fields, characterized by particular values of key soil characteristics or features, provide better areas in which to place small plots for the purpose of estimating the yield response curve. Which soil characteristics or features to use to define field areas requires further exploration, with the production sciences providing insight on which features are pertinent to the input variable being examined. However, machine learning algorithms can also be developed to identify which features to use and to discover surprising new linkages or associations between or among variables. Since these currently unknown associations are heterogeneous across fields and farms, individual farmers will benefit from collecting and using their own metadata on the features in their own fields. Furthermore, as big data becomes more prevalent in agriculture, we anticipate that machine learning algorithms will be able to sort through the large amount of soil, weather, genetic, and other information available to identify new linkages that can be used more broadly to further improve agriculture.

Fifth, recall that our algorithm also treats the number of steps or observations *T* as exogenous. In an on-farm context, this assumption implies that a farmer has a set budget to allocate for on-farm research that defines the number *T*. However, an interesting economic problem is what budget to set, as on-farm experimentation is costly but also beneficial. Costs include plot implementation and differential management, data collection and analysis, the costs for collecting the metadata for soil characteristics and other features, and costs for yield losses or

unneeded inputs for super- and sub-optimal input use on experimental plots. These costs must be balanced against the expected benefits of improved profitability in future years as a result of more efficient input use. This dynamic optimization problem is beyond the scope of this paper, but nonetheless quite interesting.

Finally, we note that the algorithm is computationally quite light and easy to run on personal computers, smartphones and devices embedded in farm machinery. We are excited about the potential for practical, low-cost, on-farm experimentation that will increase profitability, enhance sustainability, and empower individual farmers in terms of information gathering and decision making.

6 References

Athey, S. 2017. Beyond prediction: Using big data for policy problems. Science 355:483-485 DOI: 10.1126/science.aal4321.

Berry, D.A., and B. Fristedt. 1985. Bandit problems: Sequential allocation of experiments. London: Chapman and Hall.

Bubeck, S., and N. Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends in Machine Learning 5:1-122.

Coble, K., T. Griffin, M. Ahearn, S. Ferrell, J. McFadden, S. Sonka, and J. Fulton. 2016. Advancing U.S. Agricultural Competitiveness with Big Data and Agricultural Economic Market Information, Analysis, and Research. Report No 249847, C-FARE, Washington, DC.

Colquhoun, J., P.D. Mitchell, Y. Saikai. 2017. What do you really get out of all of that farm data you're collecting? Badger Commen'Tater 69(12):28-31.

Gandorfer, M., and A. Meyer-Aurich. 2017. Economic Potential of Site-Specific Fertiliser Application and Harvest Management, pp. 79-92. In S.M. Pedersen, K.M. Lind (eds.), Precision Agriculture: Technology and Economic Perspectives, Progress in Precision Agriculture DOI: 10.1007/978-3-319-68715-5_3.

Gaspar, A.P., P.D. Mitchell, and S.P. Conley. 2015. Economic risk and profitability of soybean fungicide and insecticide seed treatments at reduced seeding rates. Crop Sci 55:924-933.

Gutiérrez, B., L. Peter, T. Klein, and C. Wachinger. 2017. A Multi-Armed Bandit to Smartly Select a Training Set from Big Medical Data, pp. 38–45. In M. Descoteaux et al. (eds.), International Conference on Medical Image Computing and Computer-Assisted Intervention, Part III, LNCS 10435. DOI: 10.1007/978-3-319-66179-7_5.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. An introduction to statistical learning (Vol. 112). Springer, New York, NY.

Marra, M., D.J. Pannell, and A.A. Ghadim. 2003. The economics of risk, uncertainty and learning in the adoption of new agricultural technologies: where are we on the learning curve? Agricultural Systems 75:2015-234.

Mishra, S., D. Mishra and G.H. Santra. 2016. Applications of Machine Learning Techniques in Agricultural Crop Production: A Review Paper. Indian Journal of Science and Technology 9(38):1-14. DOI: 10.17485/ijst/2016/v9i38/95032.

Shine, P., M.D. Murphy, J. Upton, and T. Scully. 2018. Machine-learning algorithms for predicting on-farm direct water and electricity consumption on pasture based dairy farms. Computers and Electronics in Agriculture 150:74-87.

Sun, L., Y. Yang, J. Hu, D. Portery, T. Mareky and C. Hillyer. 2017. Reinforcement Learning Control for Water-Efficient Agricultural Irrigation. Ubiquitous Computing and Communications (ISPA/IUCC), 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications. DOI: 10.1109/ISPA/IUCC.2017.00203.

Sutton, R.S., and A.G. Barto. 1998. Reinforcement learning: An introduction. MIT Press, Cambridge, MA.

Algorithm 1: Reinforcement learning algorithm for selecting experimental plots

Require: *T*, field areas
Initialize: $S \leftarrow \{\ \}, K \leftarrow |\text{field areas}|, A \leftarrow \{1, 2, \dots, K\}, a_i \leftarrow 1, b_i \leftarrow 1, \forall i \in A,$
    Add two random samples to S
for $t \in \{3, 4, \dots, T\}$ **do**
    if any field area is empty **then** remove its index from A
    for $i \in A$ **do**
        Draw $\hat{q}_i$ from $Beta(a_i, b_i)$
    $j \leftarrow \text{argmax}_i\ \hat{q}_i$
    Sample (x_t, y_t) from *j*-th field area
    Add (x_t, y_t) to S and remove it from all field areas
    Estimate $\left(\hat{\alpha}_t, \hat{\beta}_t\right)$ based on S
    Compute reward r_t based on prediction $\hat{y}_t = \hat{\alpha}_t\left(1 - e^{-\hat{\beta}_t x_t}\right)$
    if $r_t = 1$ **then** $a_j \leftarrow a_j + 1$
    else $b_j \leftarrow b_j + 1$
return $\left(\hat{\alpha}_T, \hat{\beta}_T\right)$

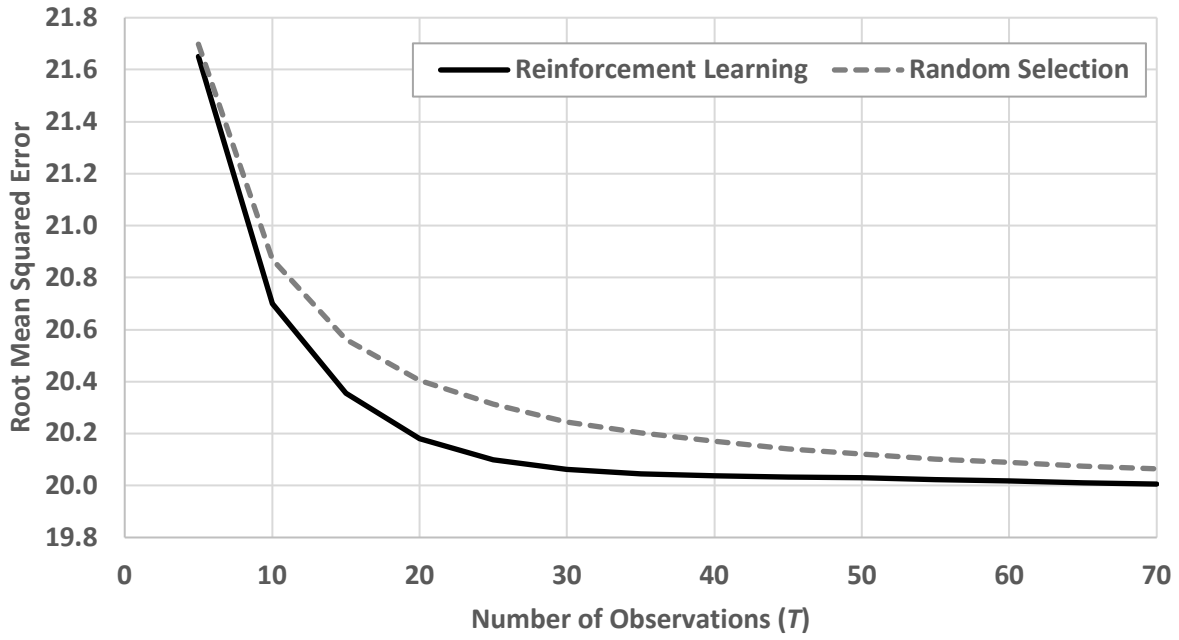

Figure 1. Root mean squared error for estimating the soybean yield response to the seeding rate using our multi-armed bandit reinforcement learning algorithm versus randomly selecting field plot observations, over a range for the number of observations used

7 Appendix

The whole procedure is implemented by the following code in Python 3.6. The input file input.csv contains four columns: "sr", "yield", "location", and "pH" for the seeding rate, soybean yield, location and soil pH data from Gaspar et al. (2015).

```
import os
import numpy as np
import pandas as pd
from scipy.optimize import curve_fit


def func(x, a, b):
  xx = np.float64(-b*x)
  return a*(1-np.exp(xx))


def algorithm1(ii,iii):
  T = TT[iii]

  rows_pool = np.array(range(num_sample))
  '''Test set'''
  rows_test     =     np.random.choice(rows_pool,     num_test,
replace=False)
  df_test = df_in.iloc[rows_test, :]
  df_test = df_test[["sr","yield"]]
  '''Validation set'''
  rows_pool = np.setdiff1d(rows_pool, rows_test)
  rows_valid     =     np.random.choice(rows_pool,     num_valid,
replace=False)
  df_valid = df_in.iloc[rows_valid, 0:2]

  '''Training set'''
  rows_train = np.setdiff1d(rows_pool, rows_valid)
  '''add two random samples to the selected set S'''
  S = pd.DataFrame(columns=df_test.columns)
  samples = np.random.choice(rows_train, 2, replace=False)
  S = S.append(df_in.iloc[samples, 0:2])
  rows_train = np.setdiff1d(rows_train, samples)
  df_train = df_in.iloc[rows_train, :]
  '''for random selection'''
  S_rnd = S.copy() # with same two samples
  samples = np.random.choice(rows_train, T-2, replace=False)
  S_rnd = S_rnd.append(df_in.iloc[samples, 0:2])

  '''Segments'''
  locations = df_train.location.unique()
  pHs = df_train.pH.unique()
  names_seg = [x for sublist in [locations, pHs] for x in sublist]
  K = len(names_seg)
  l_df_seg = list()
  for i in range(len(locations)):
    df = df_train.loc[df_train['location'] == locations[i]]
    l_df_seg.append(df.drop(["location","pH"], axis=1))
  for i in range(len(pHs)):
    df = df_train.loc[df_train['pH'] == pHs[i]]
    l_df_seg.append(df.drop(["location","pH"], axis=1))

  a = np.array([1]*K)
  b = np.array([1]*K)
  A = list(range(K))
  q = np.array([.5]*K) # arbitrary
  SSE1 = np.inf # sum of squared error
  SSE2 = 0

  for t in range(2,T):
    for i in A:
      q[i] = np.random.beta(a[i],b[i])
    arm = np.argmax(q)
    df = l_df_seg[arm]
    indices = df.index.values
    ind = np.random.choice(indices)
    S = S.append(df.loc[ind])
    for i in range(K):
      df = l_df_seg[i]
      l_df_seg[i] = df[~df.index.isin([ind])]

    '''estimation'''
    xdata = np.array(S["sr"])
    ydata = np.array(S["yield"])
    (alpha,beta), pcov = curve_fit(func, xdata, ydata, bounds=(0,
np.inf))

    '''reward'''
    r = -1
    xx = df_valid["sr"]
    yy = df_valid["yield"]
    SSE2 = sum(( yy-func(xx,alpha,beta) )**2)
    r = int(SSE2 < SSE1)
    if r==1:
      a[arm] += 1
    else:
      b[arm] += 1
    SSE1 = SSE2 # for next loop

    '''remove empty segments'''
    AA = [_ for _ in A]
    for i in AA:
      if l_df_seg[i].empty:
        A.remove(i)
        q[i] = 0

  '''random selection'''
  xdata = np.array(S_rnd["sr"])
  ydata = np.array(S_rnd["yield"])
  popt, pcov = curve_fit(func, xdata, ydata, bounds=(0, np.inf))
  alpha_rnd, beta_rnd = popt

  '''compare'''
  xx = df_test["sr"]
  yy = df_test["yield"]
  RMSE1[ii,iii] = np.sqrt(((yy-func(xx,alpha,beta))**2).mean())
  RMSE2[ii,iii]                =                np.sqrt(((yy-
func(xx,alpha_rnd,beta_rnd))**2).mean())


'''Parameters'''
TT = list(range(5,75,5)) # 5-70 w/ step 5
per_test = 0.3
per_valid = 0.2
iters = 10000


'''Main'''
print("iters =", iters)
print("TT =", TT)

os.chdir(os.path.dirname(os.path.abspath(__file__)))
df_in = pd.read_csv("input.csv")
num_sample = df_in.shape[0]
num_test = int(per_test*num_sample)
num_valid = int(per_valid*num_sample)

RMSE1 = np.zeros((iters,len(TT)))
RMSE2 = np.zeros((iters,len(TT)))
for iii in range(len(TT)):
  print("T =", TT[iii]) # progress report
  for ii in range(iters):
    np.random.seed(ii) # seed = iter number
    algorithm1(ii,iii)


'''Results'''
h = "" # for the header of output csv
for T in TT:
  h += str(T) + ","
h = h[:-1] # remove the last comma
h += ",,iters = " + str(iters)
np.savetxt("rmse1.csv",RMSE1,fmt="%.3f",header=h,delimiter=",",c
omments="")
np.savetxt("rmse2.csv",RMSE2,fmt="%.3f",header=h,delimiter=",",c
omments="")
print("RMSE1", RMSE1.mean(axis=0))
print("RMSE2", RMSE2.mean(axis=0))
```