5-7-2020

MEMS 500 Independent Study Report: Contribution Attribution via Fair Division

Kuan Lu
Washington University in St. Louis

MEMS 500 Independent Study Report: Contribution Attribution via Fair Division

Report Written by Kuan Lu

MEMS Faculty Advisor: Dr. Mark Jakiela

Abstract

This report is a summary of my work under the guidance of Dr. Mark Jakiela during the Spring 2020 semester. I have worked with Dr. Jakiela on this independent study project which complements his research interests in computer-supported collaborative work ("CSCW") systems [1]. These can be applied to higher-level, more complex tasks. This independent study examines contribution attribution via fair division. The goal of this project is to develop a program to decide whether a member in a team-based project group could receive a reward or not, based upon their contribution. I came up with the idea to use a perception learning algorithm to develop a program for this application. The report consists of four parts. The first two parts illustrate the significance of this project and why machine learning can be applied. The third and fourth parts describe how the program is developed and tested, along with an analysis of the results.

1. Introduction

This independent study is about contribution attribution via fair division. Fair division is the problem of dividing a set of resources among several people who have an entitlement to them, such that each person receives their due share. This can be modeled as an optimization problem [1]. Dr. Jakiela suggests a related concept called “contribution attribution”, which means the decision-making process that can identify the relative contribution made by each member of a team [1].

There are many examples of this problem in a student’s life. For instance, engineering students are often asked to work in teams on course projects, where all team members will get the same grades. However, some team members always complain that some members do not “pull their weight”. As another example, a group of students participate in an engineering competition with 20% of them receiving an “outstanding team member reward,”, so a fair and efficient method is necessary to decide who gets this reward.

In this independent study, a program based on a perception learning algorithm is developed, which can decide whether a student in a group project team

should receive an "outstanding group member" reward or not, based on analysis of training data. More specifically, the training data contain previous students' performance in a group project and whether they received the reward or not. After the training data are input into the program, it will study them and build a model by itself. The completed model can then decide if a student receives a reward based on their new performance. A group of test data will be imported into the program to see whether the model is accurate or not.

2. Method

Machine learning is one of the applications of artificial intelligence, focusing on the development of programs that can access data and use it to learn for themselves [2].

Machine learning is widely applied in our daily life. For instance, hospitals use machine learning to predict the probability of a patient to have a heart attack within the next year based on studying previous database. Video-sharing platforms like YouTube use machine learning to study which categories of videos you prefer, then to push advertisements based on that analysis.

The program is developed based on a perception learning algorithm. Perception learning algorithm is a machine learning method, which can find a linear separator that separates the data in D (if D is linearly separable data) within a finite number of steps [3]. For instance, banks could use perception learning algorithm to decide credit card approvals, if they have a lot of customers' information, like salary, debt, age and whether or not they defaulted on their previous credit cards. Then, a lot of computations will be applied to those historical customer data, in order to build a hypothesis model. This hypothesis model is able to decide whether to approve a credit card application or not for new customers when their personal information is collected during the application.

The goal of this independent study is using the previous data to build a model, which can evaluate the contribution of a student more reasonably and fairly, and this goal is perfectly matched with the premise of perception learning algorithm, that is "using a set of observation to uncover an underlying process" [4] [5]. How perception learning algorithm works is shown in *fig.1*.

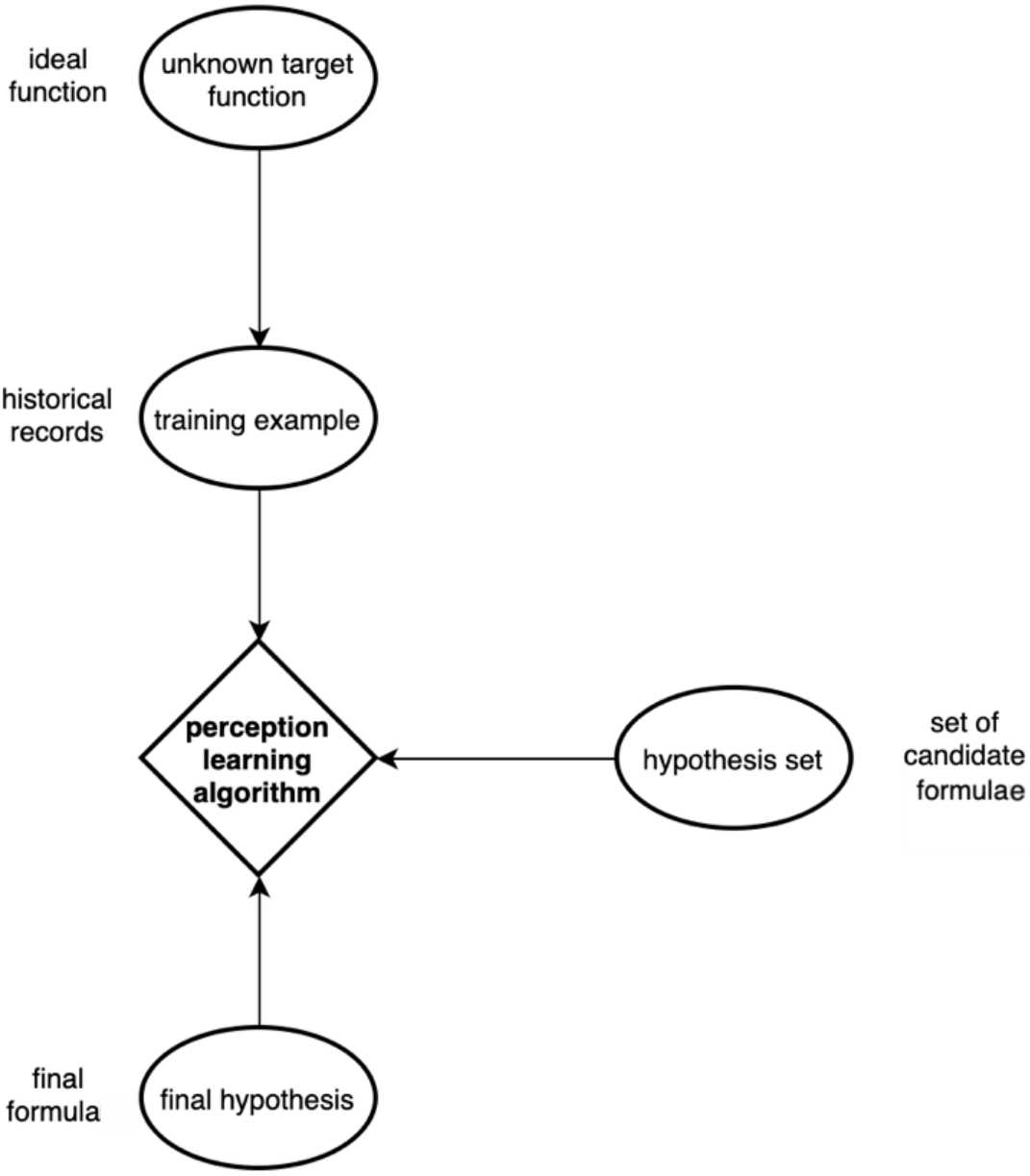

fig.1. flowchart of how perception learning algorithm works

The perception learning algorithm implements:

$$h(x) = sign(w^T x)$$

Given the training set as following:

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5) \ldots\ldots$$

and in this project, the input data set will be from previous students, specifically, x represents the performance of this student in the project, including many aspects, and will be explained in full detail later. y is a binary variable, having two values, -1 and 1, which represents whether this student receive an "outstanding group member" reward or not, respectively.

After the program self-study is completed, the algorithm produces a final hypothesis. In theory, the unknown target model is ideal and will not have any prediction error, however, the final hypothesis produced can't be exactly the same as the target model but can be close to it.

3. Program design and results

In this part, a MATLAB program based on the perception learning algorithm is developed, which can study the input data and make predictions.

The input data serves as training data, including information of a student, for instance, hours a student spent on the project, number of suggestions this student provided, and whether this student received an "outstanding group

member" reward or not. Please note that all these training data is made up by myself instead of actual collected data.

Then, a MATLAB perception learning algorithm program is built. As the name "machine learning" indicates, the program will learn from the input data and build a model by itself. After the learning process is completed, a group of test data will be imported into the program, which contains all students' information, including whether or not they received the outstanding team member reward. The program will then run the test data and predict their award decision, which will be then compared with the decision that actually happened. If the program predicts correctly most of the time, we can say that the program can successfully study the input data and make a reasonable prediction.

Perception learning algorithm will eventually converge to a linear separator for linearly separable data. Without this, the program will not converge [4].

$w(t)$ is the weights of the predicted model and w^* is the optimal set of weights, in another word, $w(t)$ will get more aligned with w^* with every iteration, assuming that $w(0) = 0$.

Let $\rho = min_{1\leq n\leq N} y_n(w^{*T}x_n)$, which is always positive because y_n and $w^{*T}x_n$ always have the same sign. The following function can be given as

$$w^T(t-1)w^* = (w_t - y_n(t)x_n(t))^T * w^*$$

So that

$$w_t^T w^* = w^T(t-1)w^* + w^*(y_n(t) * x_n(t))^T$$

We already have

$$\rho = min_{1\leq n\leq N} y_n(w^{*T}x_n) \leq y_n(t) * w^* * x_{n(t)}^T$$

Therefore,

$$w^T(t) * w^* \geq w^T(t-1) * w^* + \rho \qquad \textit{func.1}$$

The next step is to use induction to prove that

$$w^T(t) * w^* \geq t\rho \qquad \textit{func.2}$$

From *func.1*, let $t = 0$, we can have

$$w^T(0) * w^* = 0 * \rho = 0$$

And *func.2* holds. Suppose that when $t = k$, *func.1* is correct, therefore when $t = t + 1$, we have

$$w^T(k+1) * w^* \geq w^T * k * w^* + \rho \geq k\rho + p = (k+1)\rho$$

So, we can prove that *func.2* holds. Next, we have

$$\left||w(t)|\right|^2 = ||w(t-1) + y(t-1) * x(t-1)||^2$$

Which can be rewritten as

$$\left||w(t)|\right|^2 = ||w(t-1)||^2 + 2y(t-1)\left(w^T(t-1)x(t-1)\right) + ||y(t-1)x(t-1)||^2$$

Since $y(t-1)\left(w^T(t-1)x(t-1)\right) \leq 0$, we have

$$||w(t)||^2 \leq ||w(t-1)||^2 + ||y(t-1)x(t-1)||^2$$

Since $y^2(t-1)$ always be 1, we have

$$\begin{aligned} ||w(t)||^2 &\leq \left||w(t-1)|\right|^2 + ||x(t-1)||^2 \qquad \textit{func.3} \\ &\leq \left||w(t-1)|\right|^2 + max_{1<n\leq N}||x_n||^2 \\ &\leq ||w_0||^2 + max_{1<n\leq N}||x_n||^2 \end{aligned}$$

So, we have

$$||w_0||^2 + max_{1<n\leq N}||x_n||^2 = t * max_{1<n\leq N}\left||x_n|\right|^2 = t * R^2 \qquad \textit{func.4}$$

Combine *func.3* and *func.4*, we have

$$||w(t)||^2 \leq t * R^2 \qquad \textit{func.5}$$

Combine *func.2* and *func.5*, we have

$$\frac{w^T(t)w^*}{||w(t)||} \geq \frac{t\rho}{\sqrt{t} * R} = \sqrt{t} * \frac{\rho}{R}$$

Because the maximum of cos is 1, we have

$$\frac{w^T(t)w^*}{\left||w(t)|\right| * ||w^*||} \leq 1$$

This, we have

$$\frac{\rho * \sqrt{t}}{R * ||w^*||} \leq 1 \qquad \textit{func.6}$$

func.6 can be rewritten as

$$t \leq \frac{R^2 * ||w^*||^2}{\rho^2} \qquad \textit{func.7}$$

Func.7 proves that PLA will eventually converges to a linear separator for separable data, and this conclusion can be the foundation of the PLA program, acting as a theoretical basis.

The input data will be used as training data, consisting of information of 153 students. Information of the first fifty students are shown in the *tab.1* as an example. Thirteen aspects of their performances are evaluated, including:

1. Total time spent on the project (hr)
2. Whether this student participated in the model building/program writing or not
3. How many slides this student completed
4. How much time this student spent on their first slides (min)
5. How much time this student spent on their second slides (min)
6. Whether this student missed the related lecture (where the instructor described the assignment) or not
7. How many times this student consulted the professor about the project
8. How much time this student spend on their progress report (min)
9. Whether this student gave a presentation or not
10. How much time this student spent on testing the result (hr)
11. How many useful suggestions this student provided

12.How many TA help sessions this student attended

13.How many group meetings this student attended

And it will be decided whether these students could receive an “outstanding group member” reward. This is shown in the last row, with 0 meaning they don’t and 1 meaning they do.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	total hours	software?	slides num	time on slides 1	time on slides 2	miss lecture?	meet prof?	time on report 1	pre?	time on testing	advice	TA session	group meeting	decision
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
4	56	1	2	120	236	0	0	178	0	0.8	1	0	3	0
5	57	0	4	120	354	0	0	163	1	0.6	1	0	3	0
6	63	1	4	130	254	0	2	147	0	1.4	2	1	7	1
7	57	1	4	140	192	0	0	148	0	0.4	2	0	6	0
8	56	0	2	140	294	0	2	153	0	1.3	2	0	3	0
9	44	1	2	120	263	0	0	173	0	0	1	0	7	0
10	52	1	3	172	199	1	0	162	0	0.5	1	0	7	0
11	57	1	3	150	168	0	0	174	0	1.6	1	0	3	0
12	48	1	2	110	229	0	0	168	0	1	3	0	7	1
13	54	1	4	140	239	0	0	160	0	1.2	1	0	3	0
14	48	0	3	130	275	0	0	139	0	0.2	1	0	3	0
15	49	1	2	130	266	0	0	171	0	0.6	1	0	3	0
16	58	0	1	150	283	1	2	162	0	1	1	0	3	0
17	58	1	2	120	284	0	2	160	0	1.8	2	0	3	1
18	58	1	3	132	224	0	2	173	0	3.2	1	2	7	1
19	60	1	4	130	206	0	2	132	1	2.4	2	2	7	1
20	50	0	3	120	219	0	0	158	0	1.6	2	0	3	0
21	58	0	3	120	340	0	0	172	0	0	1	0	3	0
22	60	1	4	117	230	1	0	160	1	1.4	1	2	7	1
23	64	1	3	140	335	0	0	158	0	0	1	0	3	1
24	59	1	4	135	234	0	0	161	0	0.5	2	0	7	0
25	43	1	4	120	177	0	2	120	1	2.5	2	0	7	1
26	71	0	2	160	302	0	0	162	0	0.4	1	2	3	0
27	61	0	4	130	330	0	2	169	0	0	1	0	3	1
28	58	1	3	112	230	0	2	165	0	2.5	2	1	7	1
29	51	1	3	110	175	0	0	123	0	0.6	1	0	3	0
30	65	0	3	140	417	1	2	157	0	0.8	1	1	3	0
31	41	0	2	105	198	0	0	168	0	0	1	1	3	0
32	65	1	4	120	177	0	0	140	0	0.4	1	0	7	0
33	44	1	2	130	219	0	2	188	0	0	1	0	3	0
34	60	1	4	130	253	0	0	144	1	1.4	1	1	7	1
35	41	1	4	110	172	0	2	158	0	0	1	0	7	1
36	54	1	3	125	273	0	2	152	0	0.5	3	1	3	0
37	51	0	4	130	305	0	0	142	1	1.2	2	0	7	1
38	54	1	4	120	188	0	0	113	0	1.4	2	1	7	1
39	60	1	4	145	282	0	2	142	1	2.8	2	2	7	1
40	67	1	4	125	254	1	0	163	0	0.2	2	2	7	1
41	62	1	4	120	267	0	0	99	1	1.8	2	2	7	1
42	65	0	3	160	360	0	2	151	0	0.8	1	0	3	0
43	58	1	4	150	270	0	2	111	1	0.8	1	0	7	1
44	66	1	4	120	302	0	2	151	0	0.4	2	0	3	0
45	62	1	3	130	231	0	0	146	0	1.8	2	3	7	0
46	59	1	4	110	239	0	2	142	1	1.2	2	1	7	1
47	60	0	4	150	258	0	2	157	0	2.6	2	2	7	1
48	52	1	2	134	201	0	0	158	0	0.8	1	1	3	0
49	34	1	1	118	182	0	2	174	0	0	1	0	3	0
50	57	0	4	128	303	0	2	159	0	0	1	1	3	0
51	49	1	3	120	188	0	0	139	0	2	2	3	7	1

tab.1 an example of input data

The test data includes the same thirteen aspects of students' performance, and this is provided for 145 additional students. The program will generate award decisions for these 145 after studying the 153-student- training data set. then the training error will be calculated by comparing the difference between the actual and predicted awards for the 145 input students.

The MATLAB codes are shown in *fig.2a and fig.2b*, which includes two parts. A logistic regression is used to build the perception in the first part, and this part can address the training data. By integrating the training data iteratively, it will automatically generate a hypothesis model and improve it. The second part is to calculate the test error, which is between 0 and 1. This program can also compare the result difference between different numbers of iterations, $10^4, 10^5$ and 10^6 iterations are tested respectively.

```
%input training data
train_data=importdata('studenttrain.csv');
X_train=train_data.data(:,1:end-1); %152*13 the 14 column is y
y_train=train_data.data(:,end);
y_train(y_train==0)=-1; %the labels in the data set are 0/1 convert those to -1/+1

%input test data
test_data=importdata('studenttest.csv');
X_test=test_data.data(:,1:end-1); %145*13 the 14 column is y
y_test=test_data.data(:,end);
y_test(y_test==0)=-1; %the labels in the data set are 0/1 convert those to -1/+1

%part a
%pla( X, y, w_init, max_its, eta )
w_init = zeros(14,1); %add w0
max_its =[10^4,10^5,10^6];
eta=10^-5;

for i=1:3
   % t1=clock;
   tic
    [ w, e_in(i) ] = logistic_reg( X_train, y_train, w_init, max_its(i), eta );
   % t2=clock;
   % etime(t2,t1)
   toc
    train_error(i) = find_test_error( w, X_train, y_train );
    test_error(i)=find_test_error( w, X_test, y_test );
end
```

fig.2a partI of the matlab program

```
function [ test_error ] = find_test_error( w, X, y )

    [m,n] = size(X);
    X1=[ones(m,1),X];
    %Use each model to classify the data using a cutoff probability of 0.5.

    hx=exp(w'*X1')./(1+exp(w'*X1'));
    h=sign((hx)-0.5);

    %get the test_error:
    %test_error = sum(abs(h'-y)/m);

    %should be between 0 and 1 , make y to be +1 -1 so multiply 1/2
    test_error = 1/2*(sum(abs(h'-y))/m);

end
```

fig.2b partII of the matlab program

After the runs are completed, elapsed time training, and training error can be found in the workspace, which are collected and shown in *tab.2.*for our training and test sets.

Iteration (times)	10^4	10^5	10^6
Elapsed time (s)	18.6259	196.0119	1982.3024
Training error (0-1)	0.3092	0.2237	0.1513
Test error (0-1)	0.3172	0.2069	0.1310

tab.2 running result

4. Discussion

From *tab.2*, it shows that the MATLAB program can work out successfully, especially when the program iterates for 10^6 times, the test error is relatively small, only 0.1310. Therefore, the perception learning algorithm can be successfully applied to this specific simplified fair division problem, which can provide a new and more fair way to give awards.

Tab.2 also indicates a way to increase the accuracy of the predicted model,

which is to increase the maximum number of iterations. As we can see, test error decreased from more than 0.3 to 0.1310 when iterations increase 100 times, form 10^4 to 10^6.

Although a 10% error is desirable, since it is still too high for real applications, some other methods could be applied to decrease the error. For instance, the program could study a larger training data set to create a more accurate model. Also, we could pre-treat the training dataset, dealing with the missing values and detecting outliers, to get a more accurate predicted model [7].

As mentioned in the beginning, fair division will have an essential place in the future of academia and industry, providing a new way to evaluate contributions of a group work member. Many other machine learning methods could also be applied in fair division, like k-nearest neighbor. Also, there is much more that could be studied to improve the accuracy of predicted model and running speed of the program.

Reference

[1] Mark J. Jakiela. "Contribution Attribution as the Possible Next Step for “Crowdsourced” Engineering Design and Product Development." *Washington University Journal of Law & Policy* Volume 30 (2009).

[2] Porter, Bruce, and Raymond J. Mooney, eds. *Machine Learning Proceedings 1990: Proceedings of the Seventh International Conference on Machine Learning, University of Texas, Austin, Texas, June 21-23 1990*. Morgan Kaufmann, 2014.

[3] Ghahramani, Zoubin. "Probabilistic machine learning and artificial intelligence." *Nature* 521.7553 (2015): 452-459

[4] Ng, Hwee Tou, Wei Boon Goh, and Kok Leong Low. "Feature selection, perceptron learning, and a usability case study for text categorization." *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*. 1997.

[5] Stephen, I. "Perceptron-based learning algorithms." *IEEE Transactions on neural networks* 50.2 (1990): 179.

[6] Kalyanakrishnan, Shivaram. "The Perceptron Learning Algorithm and its Convergence." (2017).

[7] Acuna, Edgar, and Caroline Rodriguez. "The treatment of missing values and its effect on classifier accuracy." *Classification, clustering, and data mining applications*. Springer, Berlin, Heidelberg, 2004. 639-647.