# EstimateX

Bhavya Kedar - 201901028

SC-205 Application Project, Semester 2, DAIICT

## 1 Introduction

**One encounters the problem of estimating an outcome at every point of time. In fact, it is a very essential part of decision making. Our brain makes a decision after estimating the outcome of all possible decisions. The problems related to estimation are everywhere. A doctor has to make an estimation of what disease a patient might be suffering from based on the symptoms. Even weather forecasting consists of an estimation based on a wide range of parameters. Price of a house needs to be estimated before buying or selling, in order to avoid attaining a loss. We can estimate the volume of an object based on its dimensions as the parameters of our estimation. But sometimes it becomes difficult to estimate and make a good decision. This problem of estimating something accurately, can be made possible by using 'Matrices'.**

## 2 Main Idea

**My idea, EstimateX is about using a programmed machine to perform the process of estimation for the user. EstimateX uses 'Matrices' to perform this task. For example, let us assume that you want to sell your house that is in the main city of Ahmedabad and you are not aware of the current market price of your house. Now here, let us assume that the price of a house depends only on the size (area) of the house. If you can provide the sizes and prices of some random houses of your area that were recently sold, to the machine, then it can process over the data that you provided and generate a hypothesis (formula) that can estimate the prices of any house that is in your area if you provide its size as the input. The machine is not restricted to perform for a single parameter, it can work for any number of parameters. Expanding the same example of house prices, we can further say that the parameters for the price estimation can be size, number of floors, number of bedrooms, distance from the main city,**

**age of the house, etc. Another example can be of weather forecasting. Weather Forecasting is also an estimation process to predict the weather. It depends on a a lot of parameters such as the speed of the wind, temperature, humidity, relative pressure with the surrounding areas etc. If the machine is provided with a large training set of the examples having these parameters and their respective resulting weather, then the machine can process over the data and learn how to predict the weather accurately based on these parameters. A lot of estimation process is required to take decisions related to Share Markets. One has to decide whether to hold on to his shares or sell them in order to attain maximum profit or bare minimum loss.**

## 3 How does it work??

**Let us say you want to estimate a entity *Y* which depends on *N* number of parameters $A_1, A_2, A_3, ..., A_N$. User is supposed to provide a training set *X* containing some training examples $X_1, X_2, X_3, ..., X_M$. Here each training example contains all the specific parameters and the their respective result *Y*. Machine converts this data into matrices. Machine performs some calculations using these matrices in order to generate a hypothesis which is a polynomial whose all the variables are all the parameters $A_1, A_2, A_3, ..., A_N$ and which contains all the possible terms having power less than or equal to the 'Power of Estimation' provided by the user. The value of this polynomial when we insert the value of each parameter will be equal to the estimated value of *Y*, for those parameters.**

## 4 Whole Process of the Program

**Let us assume that the entity *Y* depends on two variables *A and B* . Now let us assume that the user has entered 5 training examples $X_1, X_2, X_3, X_4 and X_5$ and the power of the estimation is set to 2. Hence our input matrix will look like bellow :**

**Input Matrix *I* =**

$$\begin{bmatrix} X_{1,A} & X_{1,B} & Y_1 \\ X_{2,A} & X_{2,B} & Y_2 \\ X_{3,A} & X_{3,B} & Y_3 \\ X_{4,A} & X_{4,B} & Y_4 \\ X_{5,A} & X_{5,B} & Y_5 \end{bmatrix}$$

**where $X_{1,A}$ = Parameter *A* of first training example and $Y_2$ = resulting *Y* of the second training example.**

**Now, the complete polynomial of power 2 and variables *A and B* is** p $= 1 + A + B + AB + A^2 + B^2$

**Now, the machine will generate a training set *X* from our input matrix *I* based on the structure of polynomial *p* such that,**

**Training Set *X* =**

$$\begin{bmatrix} 1 & X_{1,A} & X_{1,B} & X_{1,A} * X_{1,B} & X_{1,A}^2 & X_{1,B}^2 \\ 1 & X_{1,A} & X_{1,B} & X_{1,A} * X_{1,B} & X_{1,A}^2 & X_{1,B}^2 \\ 1 & X_{1,A} & X_{1,B} & X_{1,A} * X_{1,B} & X_{1,A}^2 & X_{1,B}^2 \\ 1 & X_{1,A} & X_{1,B} & X_{1,A} * X_{1,B} & X_{1,A}^2 & X_{1,B}^2 \\ 1 & X_{1,A} & X_{1,B} & X_{1,A} * X_{1,B} & X_{1,A}^2 & X_{1,B}^2 \end{bmatrix}$$

**Now, the machine will form a matrix Y that will contain all the results of the training examples such that,**

**Matrix *Y* =**

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{bmatrix}$$

**Now, using these matrices and a normal equation, we can get a matrix *H* that will contain the coefficients of all the terms of polynomial p. Multiplying these coefficients to their respective terms in polynomial *p*, we will get our Hypothesis $h(X)$ where *X* is the set of parameters and the value of $h(X)$ will be equal to the estimated value of *Y* for that set of parameters *X*.**

**H** $= (X^T * X)^{-1} * X^T * Y$

**where $A^T$ = Transpose of A and $A^{-1}$ = Inverse of A.**

**Let's assume that *H* =**

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}$$

**This implies that our hypothesis $h(X)$ is :**

**h(X)** $= \theta_1 + \theta_2 * A + \theta_3 * B + \theta_4 * A^2 + \theta_5 * B^2$

## 5 Proof of H $= (X^T * X)^{-1} * X^T * Y$

**Assuming there to be a hypothesis matrix *H*, using linear algebra we know that *X*H* will give you a matrix that will contain the estimated values for all our training examples.**

**X*H** =

$$\begin{bmatrix} h(X_1) \\ h(X_2) \\ h(X_3) \\ h(X_4) \\ h(X_5) \end{bmatrix}$$

**Let there be a Cost Matrix *J* that contains the difference between the estimated values of the hypothesis and the actual results of the training set. Hence the value of *J* will be :**

**Cost Matrix *J* = *X*H* - *Y* =**

$$\begin{bmatrix} h(X_1) - Y_1 \\ h(X_2) - Y_2 \\ h(X_3) - Y_3 \\ h(X_4) - Y_4 \\ h(X_5) - Y_5 \end{bmatrix}$$

**For the best hypothesis, *J* should be minimum. Hence the derivative of matrix *J* with respect to matrix *H* should be matrix *O*. Therefore using matrix differentiation we will differentiate matrix *J* with respect to matrix *H* and then put the R.H.S. to be *O*.**

**Using Matrix Differentiation over the above given equation,**

$$\nabla_\theta J(H) = \nabla_\theta \tfrac{1}{2}(X * H - Y)^T * (X * H - Y)$$

$$\nabla_\theta J(H) = \nabla_\theta \tfrac{1}{2}[H^T * X^T * X * H - H^T * X^T * Y - Y^T * X * Y + Y^T * Y]$$

$$\nabla_\theta J(H) = \nabla_\theta \tfrac{1}{2} tr[H^T * X^T * X * H - H^T * X^T * Y - Y^T * X * Y + Y^T * Y]$$

$$\nabla_\theta J(H) = \nabla_\theta \tfrac{1}{2}[tr H^T * X^T * X * H - 2 * tr Y^T * X * H]$$

$$\nabla_\theta J(H) = \tfrac{1}{2}[X^T * X * H + X^T * X * H - 2 * X^T * Y]$$

$\nabla_\theta J(H) = X^T * X * H - X^T * Y$

**Now, as we know that in order to get minimum *J*, we need to put $\nabla_\theta J(H) =$ *O*.**

**Hence,**

$X^T * X * H - X^T * Y = O$

$X^T * X * H = X^T * Y$

$H = (X^T * X)^{-1} * X^T * Y$

## 6 How can this idea be a Start Up??

**As we can see this application can work for any kind of estimations depending on the training sets. With some modifications in the hypothesis, this program can also be used for classification problem. Hence depending upon the requirement, this application can be modified and be used in any specific industry. It can be used by businessmen that trade in Share Markets, by the estate agents that work with property related affairs, by the medical practitioners to have a second opinion of their diagnosis, by meteorological department to predict weather, and many more. If we can modify this idea to accept pixel color densities of an image as the parameters, then we can train the machine to distinguish between many different classes using neural networks of many such hypothesis.**

**This idea can be modified and converted into several soft-wares depending upon different requirements and can be sold or given on rent. This can generate quite a good revenue as the targeted use and the targeted users both are money-rich. Also, everyone would at least like to have a second opinion about their decisions. A programmed algorithm is also very efficient to use. Also noting down the actual outcome every time, the training sets of these soft-wares can be improved and hence the performance of the program also improves automatically.**

**Another application of this program that I discovered is that it can be used for data compression. It may sound strange but it is true. It is much much easier and occupies very less memory to store a formula that can get you a Y for any value of X, than to store many pairs of Xs and Ys. If we increase the power of estimation high enough,**

**then we will be able to generate a hypothesis(formula) that is passing through all the points provided in our training set.**

## 7 Project Summary

**This was the most interesting project I have made so far. It was quite time consuming. Programming the machine to learn by itself using Matrices was the best part of this project. As per the programming aspect, the complete application development was a big challenge, especially to write a program that can generate a polynomial of all the possible terms consisting of any number of variables and has any degree. Also the online courses of Machine Learning [4] were a great help for me to get this idea. I had to look at many resources [3] [5] across the internet to gather some knowledge about Matrix Derivation in order to prove the equation.It took me long time to find and install a java library [1] for my application that can be used to easily and efficiently work with huge matrices. Also I tried many Java libraries and found the best one [2] to insert a graph view in my application. It was the first time I used Latex to write a document.I made a YouTube video for the first time, so it gave me a chance to learn something new.**

## 8 EstimateX Android Application

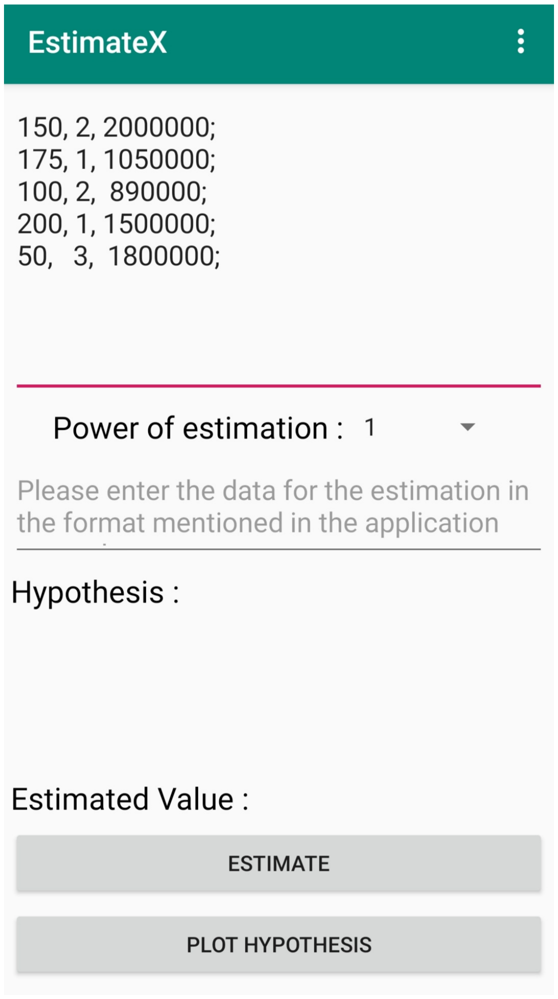
**I have also implement this idea in the from of an android application. This application accepts a training set from the user in text format and then processes the input data as show above. After the process it generates a hypothesis that can be used for estimation. User can also provide a set of parameters to be inserted in the hypothesis and the application will provide user with the estimated value for those parameters.**

**For example as mentioned above you want to estimate the price of your house but this time let us assume that the price depends upon two parameters the size (area) of the house as well as the number of floors. Assuming you have a training set of 5 training examples to perform this estimation. You need to provide your input in the following format to the application.**

$$\begin{array}{lll} Hsize_1, & Hfloors_1, & Hprice_1; \\ Hsize_2, & Hfloors_2, & Hprice_2; \\ Hsize_3, & Hfloors_3, & Hprice_3; \\ Hsize_4, & Hfloors_4, & Hprice_4; \\ Hsize_5, & Hfloors_5, & Hprice_5; \end{array}$$

**For example,**

```
150,  2,  2000000;
175,  1,  1050000;
100,  2,   890000;
200,  1,  1500000;
 50,  3,  1800000;
```



```
150, 2, 2000000;
175, 1, 1050000;
100, 2,  890000;
200, 1, 1500000;
50,   3,  1800000;
```

Power of estimation : 1

Please enter the data for the estimation in the format mentioned in the application
Hypothesis :
Estimated Value :
ESTIMATE
PLOT HYPOTHESIS

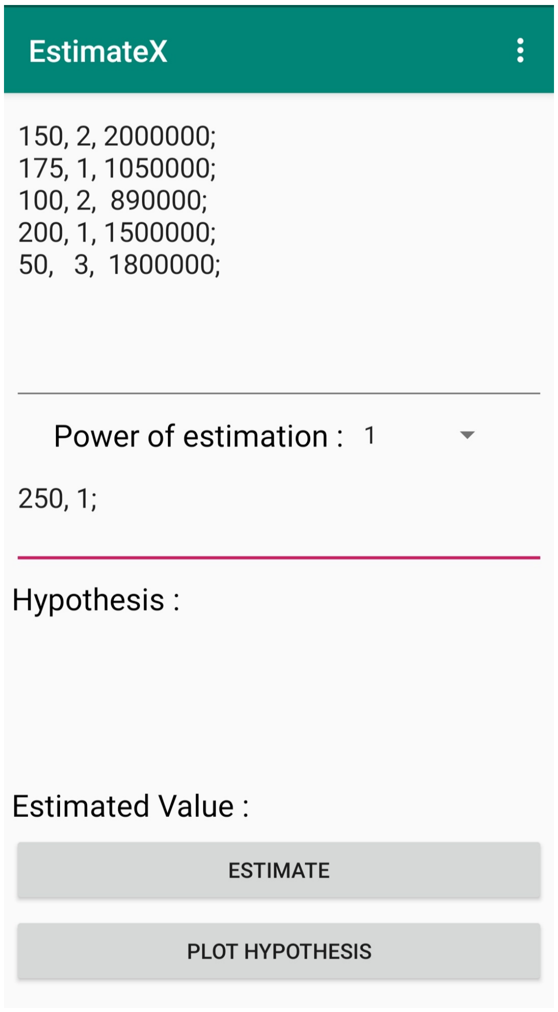
**After you have provided the training set to the application, you need to provide a set of parameters (your query) to the application if any, in the same format that we followed in the training set. Also you need to select the power of estimation. I have selected it to be 1. You can select any power from 1 to 5. NOTE : Higher power gives rise to the problem of 'Overfitting'. Because of this problem, the hypothesis will**

**give very accurate output for the training sets, but it will not give appropriate output for other queries.**

$$Hsize, \quad Hfloor;$$

**For example,**

$$250, \quad 1;$$

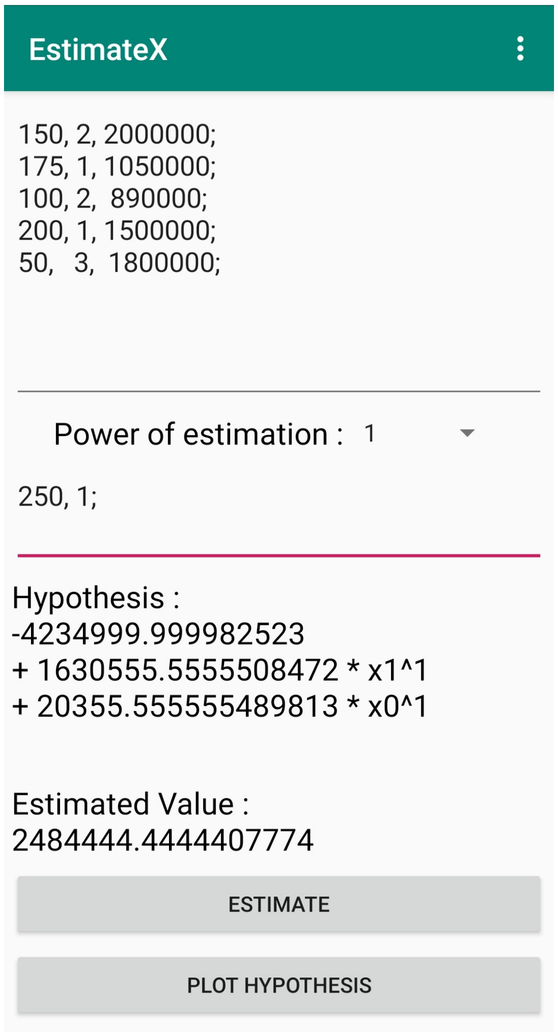


150, 2, 2000000;
175, 1, 1050000;
100, 2, 890000;
200, 1, 1500000;
50, 3, 1800000;

Power of estimation : 1

250, 1;

**After you have provided all the inputs to the application, you will have to click the Estimate Button in order to generate a hypothesis. As soon as you click on the Estimate Button, you will see the hypothesis polynomial as well as the estimated value of the query on your screen.**



```
150, 2, 2000000;
175, 1, 1050000;
100, 2,  890000;
200, 1, 1500000;
50,   3,  1800000;
```

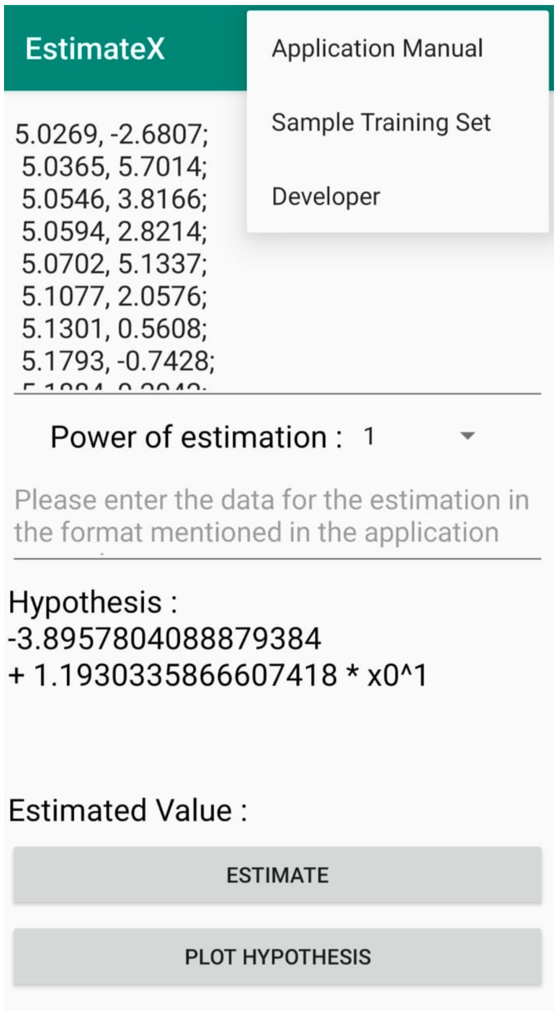
Power of estimation : 1

```
250, 1;
```

```
Hypothesis :
-4234999.999982523
+ 1630555.5555508472 * x1^1
+ 20355.555555489813 * x0^1

Estimated Value :
2484444.4444407774
```

Here we have used two parameters, but if you use only one parameter to estimate the value you can also plot the graph of the hypothesis on a Cartesian plane. I have also provided a sample training set that you can set as your input from the menu button of the application.



5.0269, -2.6807;
5.0365, 5.7014;
5.0546, 3.8166;
5.0594, 2.8214;
5.0702, 5.1337;
5.1077, 2.0576;
5.1301, 0.5608;
5.1793, -0.7428;

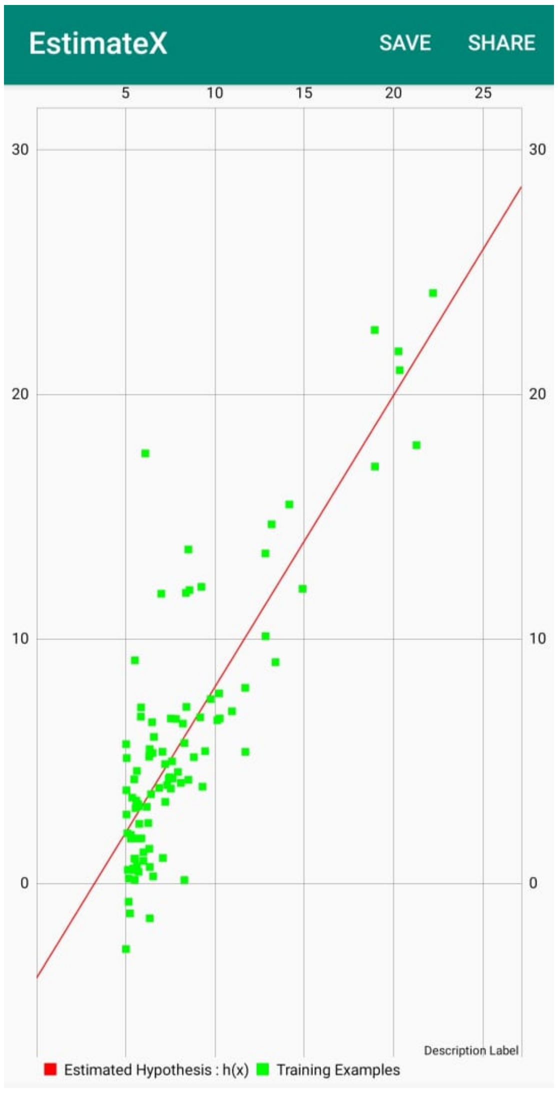
Power of estimation : 1

Please enter the data for the estimation in the format mentioned in the application

Hypothesis :
-3.8957804088879384
+ 1.1930335866607418 * x0^1

Estimated Value :

**After using the sample training set or any other single parameterized training set and estimating its hypothesis, you can use the plot hypothesis feature of the application using the Plot Hypothesis button.**



**Link to download this application :** ***Click here to download.***

**Contribute :** ***Click here to contribute.***

## References

[1] Effective java matrix library (ejml). `http://ejml.org/wiki/index.php?title=Main_Page`.

[2] Mpandroid chart, java library to use graphview. `https://github.com/PhilJay/MPAndroidChart`.

[3] Randal J. Barnes. Matrix differentation. `https://atmos.washington.edu/~dennis/MatrixCalculus.pdf`.

[4] Andrew Ng. Coursera machine learning, stanford university. `https://www.coursera.org/learn/machine-learning`.

[5] Andrew Ng. Cs229 lecture notes. `http://cs229.stanford.edu/notes/cs229-notes1.pdf`.