

# Classification of Dating Agency Participant using Support Vector Machine

Ramos Janoah Hasudungan/13514089<sup>1</sup>

Computer Science/Informatics

School of Electrical Engineering and Informatics

Institut Teknologi Bandung, Jl. Ganesha 10, Bandung, 40132, Indonesia

<sup>1</sup>13514089@std.stei.itb.ac.id

*Interest to opposite sex person surely has some factor. If we had data, we can analyzed which factor is strengthen or weaken the interest. We can analyze it with machine learning such as support vector machine to make some model to classify a data of some person, and decide whether they are classified as an attractive person or not.*

**Keywords**—Interest of opposite sex; Machine learning; Support vector machine;

## I. INTRODUCTION

Long ago, we can only find data in specific field, such as finance or sales and purchase transaction. But now, data is everywhere, because now information technology become more advance than before, and that makes data become more vary. One of them is data that gathered in Television Industry.

'Take me out' is a Reality TV shows like dating agency that help their participant found their 'mates' faster. They surveyed their participant about person that they met in the show. In that survey, they fill their assessment against the person they met, such as their physical attractiveness, intelligence, or shared hobbies, and decision that would him or her match with that person. With this kind of data, we can make a model and classification through this data, and make a prediction about what makes someone interested to another.

Support Vector Machine is one of the machine learning that can be used to make a model of classification and prediction, and it will be used in this paper.

## II. BASIC THEORY

Support vector machine (SVM) is a supervised machine learning that can be use either for classification or regression. The common use of SVM is for classification. The idea of SVM is to find the hyperplane that best divides a dataset into two classes, as shown in the image below.

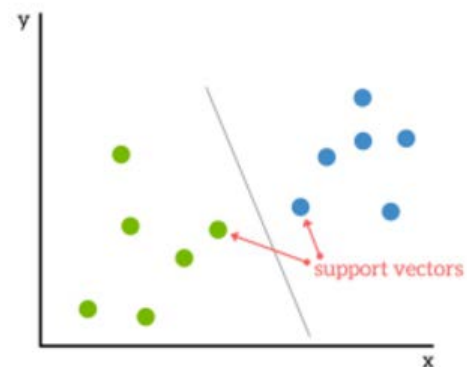


Figure 1 Example of hyperplane and support vector

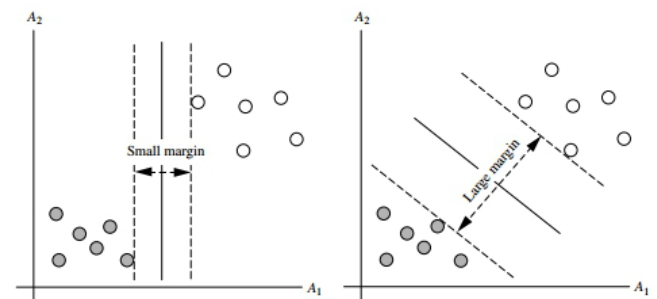


Figure 2 The comparasion of two hyperplane

The key of the support vector machine is to find the hyperplane. The example of hyperplane can be seen in the figure 1. The best hyperplane is the hyperplane that split two classes with the greatest margin with the closest vector to another class vector.[1] Just like figure 2, the right figure has a better hyperplane than the left figure. That vector that has closest range with the hyperplane would be named 'Support Vector'. By looking for the best hyperplane, hopefully it would split the prediction data better than another hyperplane[2].

Many datasets also cannot be well splited by a linear hyperplane, but can be splited better with another way. For example, how can we linearly splited data in XOR problem? To do it, we must use 'kernel trick' [3], or change the kernel of the SVM to make the better classification. Kernel is a function that we use to find the hyperplane. If we use 'linear kernel', it means that we use linear hyperplane.

The type of kernel that commonly used for Support Vector Machine is:

- Linear  
Linear kernel is just like figure 1 and figure 2. We use linear hyperplane to split the vector
- Polynomial  
Polynomial kernel is a function that split vector with a polynomial function, just like figure 3.

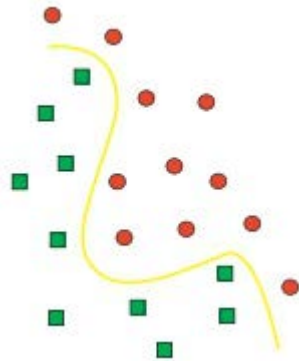


Figure 3 Example of Polynomial Kernel

- Radial  
Radial kernel is a function that split classes by the area that made radially. The example of Radial Kernel is like figure 4.

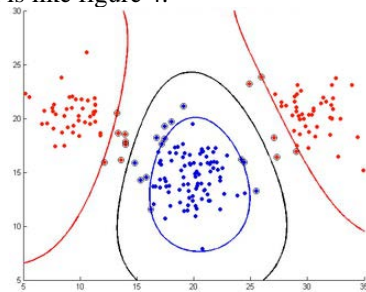


Figure 4 Example of Radial Kernel

- Sigmoid  
Sigmoid kernel function has a formula:  
$$y = 1 / (1 + e^{-x})$$

Sigmoid function would have a plot result like figure 5.

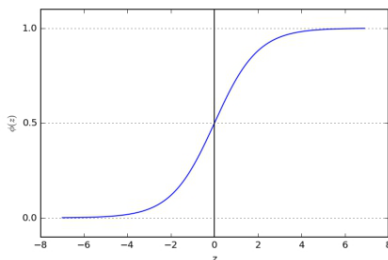


Figure 5 Example of Sigmoid Kernel

Changing the type of kernel of the SVM would make the model better to do a classification and prediction.

### III. IMPLEMENTATION

This analysis will be done with R Programming Language, and RStudio as an IDE.

Steps taken to do the data analysis is:

#### 1. Importing the data

The Data was obtained in [www.kaggle.com](http://www.kaggle.com), a website that provides a public data. The Data composed by 8378 row and 195 variables. In this data, there are so much attributes, and in this observation, author will only take some variables to be analyzed. The process of the importing data can be seen in these code.

```
# Importing Data
library(readr)
Speed_Dating_Data <- read_csv("C:/Speed
Dating Data.csv")
```

#### 2. Data cleaning

Data cleaning is process of detecting and handling inaccurate data, and also removing something in data that is unnecessary on the analysis process. What it does in this cleaning is removing the unnecessary column, until that left in data is:

- a. *iid* (unique for every person), the primary key of the respondent. The data type is integer.
- b. *gender* of the respondent (0 for male, 1 for female),
- c. The six attributes rate of their mate (Attractiveness, Intelligence, Fun, Ambitious, Sharable Interest/Hobbies, and Sincere). The rate would be represent by an integer between 0 and 10.
- d. The decision of the respondent, which is match or not match (1 for match, 0 for not match) with their mate.

After that, we split the data to which is gender is male and gender is female. This is done for analyzing data separately between male and female. The data cleaning process can be seen on these code.

```
# Data Cleaning

df = Speed_Dating_Data

# Remove all null, and remove
# unnecessary column

CleanData = na.omit(data.frame(df$id,
df$gender, df$match, df$attr, df$sinc,
df$intel, df$fun, df$amb, df$shar))

...
```

```

...
# Separate Data to Gender 0 and Gender 1
CD_0 = subset(CleanData,
              df.gender == 0)
CD_0_o = data.frame
(match = CD_0$df.match,
 att = CD_0$df.att,
 sin = CD_0$df.sin,
 int = CD_0$df.int,
 fun = CD_0$df.fun,
 amb = CD_0$df.amb,
 sha = CD_0$df.sha)

CD_1 = subset(CleanData, df.gender == 1)
CD_1_o = data.frame
(match = CD_1$df.match,
 att = CD_1$df.att,
 sin = CD_1$df.sin,
 int = CD_1$df.int,
 fun = CD_1$df.fun,
 amb = CD_1$df.amb,
 sha = CD_1$df.sha)

```

Here is the glimpse of the data now.

	match	att	sin	int	fun	amb	sha
1	0	6	9	7	7	6	5
2	0	7	8	7	8	5	6
3	1	5	8	9	8	5	7
4	1	7	6	8	7	6	8
5	1	5	6	7	7	6	6
6	0	4	9	7	4	6	4
7	0	7	6	7	4	6	7
8	0	4	9	7	6	5	6
9	1	7	6	8	9	8	8
10	0	5	6	6	8	10	8
11	0	5	7	8	4	6	3
12	0	8	5	6	6	9	6
13	0	5	8	9	6	3	4

Before using SVM, we should make the match column data type become string or character, because SVM that would be use can be use to make a classification on regression. If we keep the match column integer, then the SVM would make the regression instead of classification. Therefore, we would make 1 become 'm' character (m for match) and 0 become 'n' character (n for not match). These code to make that change.

```

# Make change for the gender 0 data
CD_0_a = CD_0_o
CD_0_a$match[CD_0_a$match == 1] <- "m"
CD_0_a$match[CD_0_a$match == 0] <- "n"

# Make change for the gender 1 data
CD_1_a = CD_1_o
CD_1_a$match[CD_1_a$match == 1] <- "m"
CD_1_a$match[CD_1_a$match == 0] <- "n"

```

### 3. Using the SVM to make a model

Now the preparation to make a model is set up. To

make the model, we would use package e1071. This package is made for R Programming Language that has several machine learning that ready to be used. One of them is Support Vector Machine.

These code are the process of making the model.

```

# Use the library
library(e1071)

# SVM
# Gender 0
model_G0 <- svm(CD_0_a$match~.,
                data=CD_0_a, kernel = "linear", type =
                "C-classification")
summary(model_G0)

# Gender 1
model_G1 <- svm(CD_1_a$match~.,
                data=CD_1_a, kernel = "linear", type =
                "C-classification")
summary(model_G1)

```

These are the example output of the summary

```

# Gender 0
Call:
svm(formula = CD_0_a$match ~ ., data =
CD_0_a, kernel = "linear", type = "C-
classification")

Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  linear
      cost:  1
      gamma: 0.1666667

Number of Support Vectors: 1253

( 662 591 )

Number of Classes: 2

Levels:
 m n

```

### 4. Measure the SVM performance.

The performance of classification can be measured by comparing the accuracy of the prediction. The accuracy formula is total of record that classified right divided by total record that classified. Here are the example of the accuracy measurement, using confusion matrix and accuracy formula.

```

> model_G0 <- svm(CD_0_a$match~., data=CD_0_a,
+               kernel = "sigmoid",
+               type = "C-classification")
> pred <- predict(model_G0, CD_0_a)
> table(Predicted = pred, Actual = CD_0_a$match)
      Actual
Predicted m  n
m      142 428
n      449 2438
> cat("Accuracy of classification : ",
+     sum(diag(conf_matrix))/sum(conf_matrix))
Accuracy of classification : 0.8290425

```

Now, we can compare all the kernel function to find the better kernel function. Here are the comparison table.

Kernel function	Accuracy in Gender 0	Accuracy in Gender 1
Linear	0.829425	0.820022
Polynomial	0.829425	0.820022
Radial	0.845521	0.836024
Sigmoid	0.849233	0.844872

We can see that the accuracy is slightly better when we use sigmoid function. This indicates that for this case, sigmoid is a better kernel function to do classification.

#### IV. CONCLUSION

Support Vector Machine can be used to do classification. It has a good enough accuracy to do classification, but there are more machine learning that developed that could give a better result than Support Vector Machine, such as artificial neural network.

#### V. ACKNOWLEDGMENT

I would like to thank the lectures of IF3280 Socio-Informatics and Professionalism course, Dr.Ir. Rinaldi Munir, MT., Dr. Eng. Ayu Purwarianti, ST., MT. and Dr. Dessi Puji Lestari, who encourage author to learning new things, especially for this data classification, that could be helpful for the author to learn something new to become data scientist.

#### REFERENCES

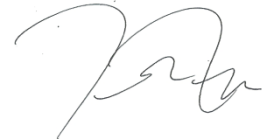
- [1] Bambrick, Noel. Support Vector Machines: A Simple Explanation. Retrieved May 1, 2017, from <http://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html>
- [2] I. H. Witten et al., Data Mining, 3rd ed. Burlington: Morgan Kaufmann, 2011.
- [3] Kim, Eric. Everything You Wanted to Know about the Kernel Trick. Retrieved May 3, 2017, from [http://www.eric-kim.net/eric-kim-net/posts/1/kernel\\_trick.html](http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html)

#### STATEMENT

This Statement show that I agree that what I wrote is my own writing, not plagiarism or translation from others paper

Bandung, 5 May 2017

Signature



Ramos Janoah Hasudungan/13514089