A Logistic Regression Approach to Key Factor Identification of Forming Social Circles

Team: 7

R09942069 蕭如芸 R09942089 黃郁珊 R09921097 張珮萱 R09921088 李昕耘

1. Introduction

A social circle is a group of socially connected people, such as a group of close friends, family members or colleagues. In this project, we utilize the dataset on Kaggle: Learning Social Circles in Networks [1]. The dataset is collected by Facebook, which models friend memberships to multiple circles. In this project, we aim to find out the key factors (i.e., features) that form each circle. Here, we introduce the data used in our task.

- `featureList.txt`: List of all possible features, a total of 57.
  - e.g., birthday, education;type, first_name, gender, hometown;name, etc.
- `features.txt`: Contains features for all users.
  - Each line is of the form
    `UserId feature1;value1 feature2;value2 feature3;value3 ...`
  - The feature values have been hashed to a number and thus anonymized. For example, a line of the form
    `20899 first_name;5962 last_name;13230 name;20731 locale;5`
    means that user "20899" has `first_name` "5962", `last_name` "13230", etc.
- `Training/*.circles`: Each `user.circles` file contains all circles to which `user` belongs.
  - Each line is of the form
    `circleID: friend1 friend2 friend3 ...`
    This means that `user` belongs to `circleID`, and so do `friend1`, `friend2`, `friend3`, etc.

2. Related Work

We refer to the paper "Learning to Discover Social Circles in Ego Networks" [2]. In this paper, the goal is to identify users' social circles in an unsupervised learning method. The likelihood of forming a circle can be calculated by the following equation.

$$d_k(e) = \delta(e \in C_k) - \alpha_k \delta(e \notin C_k), \qquad \Phi(e) = \sum_{C_k \in C} d_k(e) \langle \phi(e), \theta_k \rangle ,$$

where e is the edge that connects two users, $\phi(\cdot)$ is a similarity function which computes the similarity of each feature of an edge (between two users), and θ_k is a parameter to be learned, which encodes how the circle C_k emerged. To be specific, θ_k is a weight vector representing the importance of each feature.

The objective is to maximize $\Phi(e)$ by learning an optimal θ_k for each circle C_k. The idea is that if e belongs to C_k, $\langle \phi(e), \theta_k \rangle$ should be large. If e does not belong to C_k, $\langle \phi(e), \theta_k \rangle$ should be small.

Inspired by the above relation, we know that we can obtain important features by learning the weights of each feature (i.e., θ_k). In addition, we can learn θ_k by optimizing the **inner product** of the similarity of each feature between two users (i.e., $\phi(e)$) and the weights of each feature (i.e., θ_k).

3. Proposed Method

3.1 Problem Formulation

For each circle C_k, we aim to find the corresponding θ_k that represents how each circle forms. θ_k is a vector in which each element represents the weight of the corresponding feature. Features corresponding to higher weights are more important features that form the circle. The following is an overview of the implementation process.

Step 1: As illustrated in Figure 1, for each circle, sample two positive samples (x_1 and x_2) and one negative sample ($\bar{x}$).

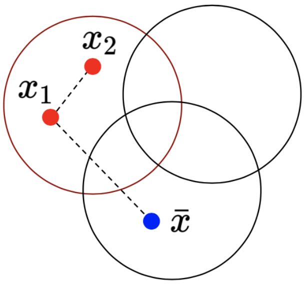


Figure 1: Schematic diagram of sampling data. The red points are positive samples, while the blue point is the negative sample.

Step 2: Compute the similarity function $\phi(x_1, x_2)$ and $\phi(x_1, \bar{x})$.
The output of the similarity function ϕ is a vector of dimension n, where n is the

number of features. Take $\phi(x_1, x_2)$ as an example, if the values of feature i in x_1 and x_2 have one in common, we set the similarity to 1, and set to 0 if no feature values are in common.

Step 3: Model the task as **logistic regression**.
We model the task as a linear transformation

$$y = wx + b,$$

where x is the similarity of each feature between two samples (i.e., $\phi(x_1, x_2)$ or $\phi(x_1, \bar{x})$), w is the weight of each feature (i.e., θ), y is the class to predict, and that is, whether the two samples are both in this circle (1 if both belong to this circle, and 0 otherwise). Calculating the inner product of the similarity $\phi(\cdot)$ and the feature weight θ follows the reference paper introduced in section 2.

Our goal is to find an optimal w for each circle. That is, find out the key common features that form the circle. The implementation details and how we apply logistic regression will be detailed in the following subsections.

3.2 Data Pre-processing

By observing the feature list, we found that some features are not very meaningful. In particular, we believe that work;start_date, work;end_date, work;projects;start_date and work;projects;end_date are less meaningful features. Instead, whether the work dates of two individuals "**overlap**" has more influence on whether the two are in the same circle. However, we cannot know whether the work dates overlap directly from the given datasets, since the feature values are hashed to meaningless numbers. Therefore, it requires some data pre-processing.

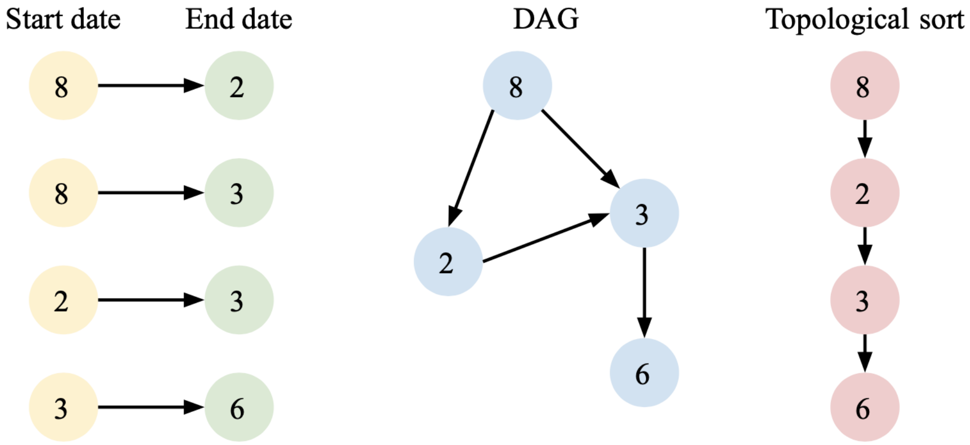


Figure 2: Illustration of the directed acyclic graph and the topological sort.

As illustrated in Figure 2, we model the "start dates" and "end dates" of all users as a directed acyclic graph (DAG), where each "start date → end date" forms an edge. Then, we apply the "**topological sort**" [3] to the graph, which transforms the directed graph to a linear ordering of its vertices such that for every directed edge uv from vertex u to vertex v, u comes before v in the ordering.

Then, given the start dates and end dates of two individuals, we can know whether their work dates overlap from the ordered list generated by the topological sort. The "overlap" can be treated as a new feature.

3.3 Logistic Regression

Logistic regression models the probabilities for classification problems with two possible outcomes. Therefore, it can be utilized to our task, predicting whether two samples belong to the same circle. The process can be divided into 3 steps.

Step 1: Define the function set.
The output is modeled as a function $f(x)$, which is calculated by

$$f(x) = \sigma(wx + b),$$

where x is the input. x will first undergo a linear transformation, where w and b are the weight and bias, respectively. Then, the transformed result goes through an activation function σ, where σ is the sigmoid function. The definition of the sigmoid function is

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

Figure 3 shows the output value with respect to different x. It can be seen that the output range is between 0 and 1. With this property, if the output value is closer to 0, we can assign it to class 0; if the output value is closer to 1, we can assign it to class 1.

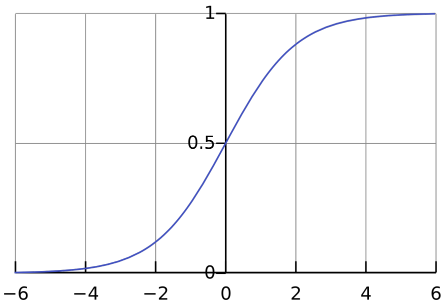


Figure 3: The sigmoid function. The horizontal axis is the value of x, while the vertical axis is the value of $\sigma(x)$.

Step 2: Evaluate the function.

In this step, we aim to evaluate the function defined in Step 1. Since we are addressing the classification tasks, the evaluation criterion should be the classification ability of the function. That is, we evaluate the function by measuring its classification accuracy. The most commonly used loss function for classification is the cross entropy loss, defined as

$$L = -\left[y\ln f(x) + (1-y)\ln\left(1 - f(x)\right)\right],$$

where y is the ground truth label (i.e., 0 or 1).

Step 3: Find the best function.

In this step, the goal is to find the best function. To be specific, we find the best parameters w and b that minimize the loss defined in Step 2. To solve this optimization problem, we apply the **gradient descent** method. The procedure will be detailed in the following subsection.

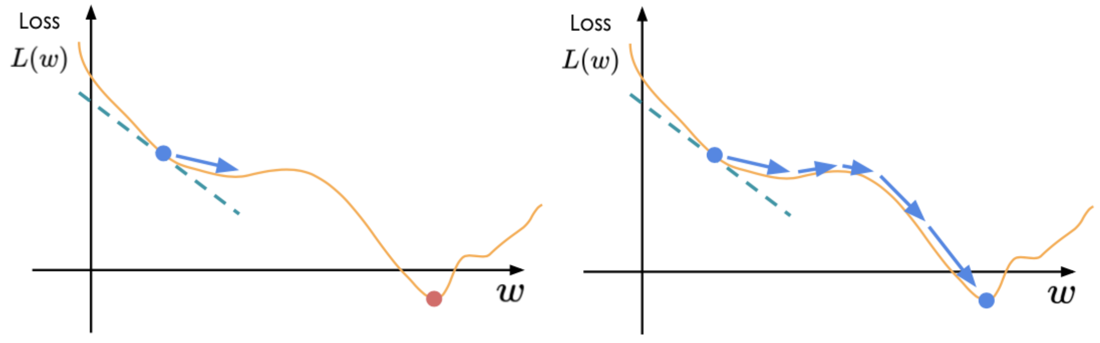
3.4 Gradient Descent

Gradient descent is an iterative optimization algorithm for finding a local minimum of a differentiable function. Therefore, we can apply gradient descent to find the best parameters that minimize a differentiable loss function. An example is illustrated in Figure 4. The orange line is the loss curve corresponding to different w values. As shown in Figure 4(a), the blue point represents the initial position. Our goal is to find the optimal red point. Since the loss function may be complicated, we cannot find the optimal solution directly. However, we can move the blue point step by step toward the red point by applying gradient descent.

Particularly, we use the concept of differentiation. At the initial point, we differentiate the loss function with respect to w, obtaining the differential value, which is also the "slope" at the blue point. Here, we can make two observations. Firstly, the larger the differential value, the larger the step it can take. Secondly, the slope is negative and the point should move to the right, and that is, increase the value of w. On the contrary, if the slope is positive, the value of w should be decreased. To satisfy the above observations, the formula for updating w can be designed as

$$w \leftarrow w - \eta \frac{dL}{dw},$$

where η is a constant known as learning rate, which is a tunable parameter that determines the step size at each iteration. After many iterations, we expect that the blue point can move to the red point (a local minimum), as shown in Figure 4(b).



(a) (b)

Figure 4: Example for gradient descent. (a) is the initial condition, and (b) is the expected final result.

Combining the concept of logistic regression and gradient descent, we can implement the optimization problem in 3 steps. The following is a brief description.

Step 1: Define function set $f(x) = \sigma(wx + b)$.

Step 2: Define loss function $CE(f(x), y)$, where CE denotes cross entropy.

Step 3: Find the best function by gradient descent

$$w \leftarrow w - \eta\big(-(y - f(x))x\big),$$

where $(-(y - f(x))x)$ is the gradient of the loss function.

The following is the calculation process of the cross entropy loss gradient.

$$f(x) = \sigma(z) = \sigma(wx + b)$$

$$\frac{\partial L}{\partial w} = -\left[y \boxed{\frac{\partial \ln f(x)}{\partial w}} + (1-y)\boxed{\frac{\partial \ln(1-f(x))}{\partial w}}\right]$$

$$\frac{\partial \ln f(x)}{\partial w} = \frac{\partial \ln f(x)}{\partial z}\frac{\partial z}{\partial w} \qquad \frac{\partial z}{\partial w} = x$$

$$\frac{\partial \ln \sigma(z)}{\partial z} = \frac{1}{\sigma(z)}\frac{\partial \sigma(z)}{\partial z} = \frac{1}{\sigma(z)}\sigma(z)(1-\sigma(z)) = 1 - \sigma(z)$$

$$\frac{\partial \ln(1-f(x))}{\partial w} = \frac{\partial \ln(1-f(x))}{\partial z}\frac{\partial z}{\partial w} \qquad \frac{\partial z}{\partial w} = x$$

$$\frac{\partial \ln(1-\sigma(z))}{\partial z} = -\frac{1}{1-\sigma(z)}\frac{\partial \sigma(z)}{\partial z} = -\frac{1}{1-\sigma(z)}\sigma(z)(1-\sigma(z)) = -\sigma(z)$$

$$\begin{aligned}\frac{\partial L}{\partial w} &= -[y(1-f(x))x - (1-y)f(x)x] \\ &= -[y - yf(x) - f(x) + yf(x)]x \\ &= \underline{-(y - f(x))x}\end{aligned}$$

3.5 Data Post-processing

We first select the top 3 most weighted features as representation for each circle. However, the result turns out to be noisy and contains redundant information, and thus it is difficult to divide the circles into several main categories to perform further analysis. Therefore, we perform two types of data post-processing to obtain “cleaner” important features of each circle.

Setting a threshold

Of the top 3 most weighted features, eliminate the features with weight less than 1.0.

Computing the correlation between features

Of the top 3 most weighted features, keep only part or all of them so that the correlation of each pair is less than 0.2. This can remove the redundant features.

4. Results

4.1 Effect of Data Pre-processing

Take circle 16 as an example, the top 3 most weighted features are “last_name”, “overlap” and “location;id”. In this circle, the feature “overlap” weighs 2.0603, while “work;end_date” and “work;start_date” weigh 0.5683 and 0, respectively. It shows that “overlap” can indeed act as an important feature of circle grouping. This can also be observed in other circles (e.g., circle 13, 650).

4.2 Effect of Data Post-processing

From the calculation result of correlation, we found that “id” and “name” are highly correlated. We guess that they may be different representations of the same thing. For example, the following tuples have extremely high correlation:

(work;projects;from;id, work;projects;from;name)
(work;from;id, work;from;name)
(work;location;id, work;location;name)

The purpose of data post-processing is to obtain cleaner data for analysis. We compare the number of circle types before and after data post-processing. Note that if two circles have the same “most weighted features”, they are considered the same type of circle. A circle type is represented by at most 3 features, as described in subsection 3.5. As shown in Table 1, after data post-processing, the number of circle types has been reduced by **92**.

Table 1: Number of circles and number of circle types.

Total number of circles		455
Number of circle types	Before data post-processing	249
	After data post-processing	157

4.3 Quantitative Results

In Figure 5, we calculate the number of occurrences of each feature in all circle types. We can see that about half of the features have very low influence on the formation of the circle. Moreover, there are quite a few different types of features appearing more than 10 times, indicating that there are indeed various types of circles.

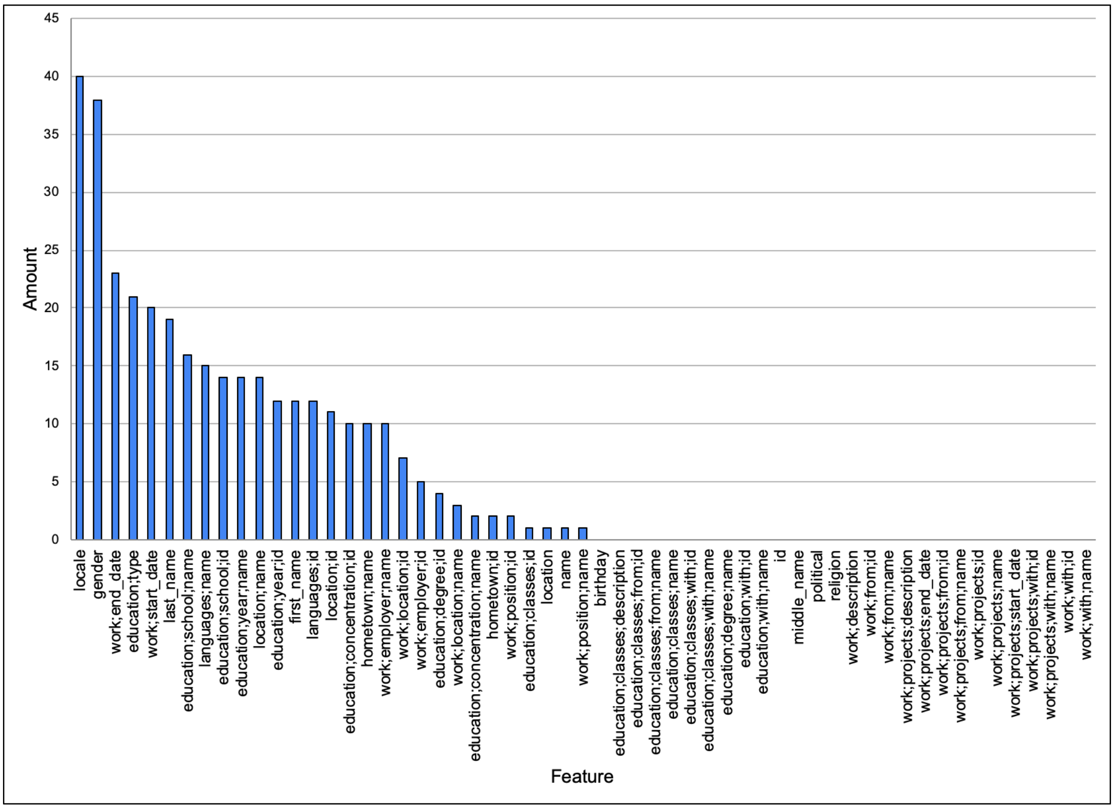


Figure 5: According to our method, the number of occurrences of each feature in all circle types.

4.4 Comparison with Baseline Method

We conduct a statistical method as baseline. For all pairs of users belonging to the same circle, calculate whether they have a common feature. As a result, we can obtain the number of times each feature has a common feature value among users belonging to the same circle. The result is presented in Figure 6. We can see that the top 3 weighted features are “locale”, “gender” and “education;type”, which do not differ much from the result of our method. However, when it comes to a single circle, the result turns out to be much more meaningful.

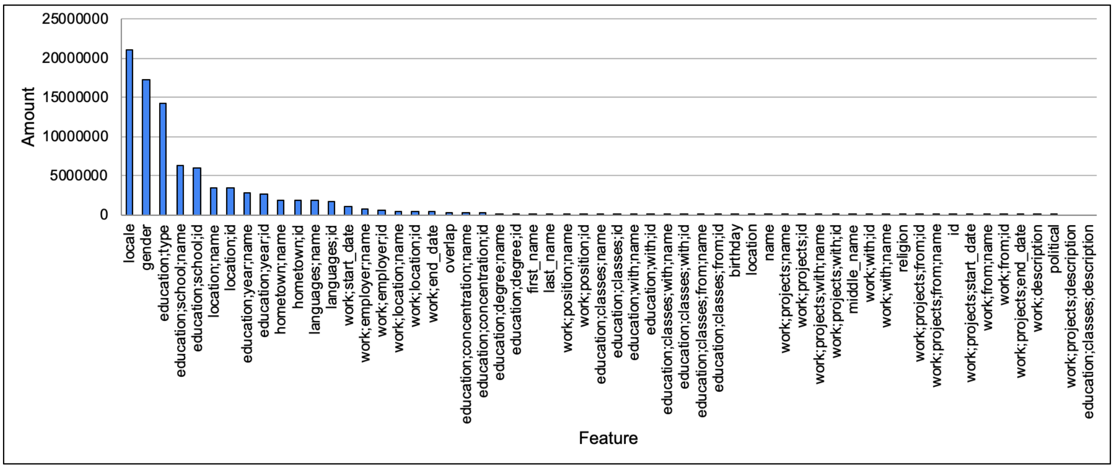


Figure 6: The number of occurrences of each feature in all circles obtained by a statistical method (baseline).

To make further explanation, we select some circles to demonstrate the difference between the baseline method and our method. Note that the statistical results of baseline here are calculated based on a single circle instead of all circles. Table 2 lists the top 3 features obtained by the baseline method and our method. It can be seen that the results of the baseline method are all “locale”, “gender” and “education;type”, just the same as the overall statistical result. That is, it fails to distinguish the unique key factors of each circle and is heavily affected by the redundant features.

On the contrary, our method is capable of overcoming this obstacle. Table 2 shows that our results are diverse, and that is, our method has the ability to vary depending on different types of social circles. We believe the results obtained by our method are more convincing and meaningful.

Table 2: Example of top 3 features obtained by the baseline method and our method.

Circle ID	Method	Feature #1	Feature #2	Feature #3
118	Baseline	locale	gender	education;type
	Ours	locale	languages;id	hometown;id
643	Baseline	gender	locale	education;type
	Ours	education;school;id	education;year;name	hometown;id
877	Baseline	gender	locale	education;type
	Ours	locale	gender	education;type

4.5 Illustration of Users Belonging to Different Types of Circles

In this subsection, we demonstrate how users belong to different types of circles. Take user 2895 as an example, he belongs to circle 10, 18, 276 and 278. Table 3 lists the circle types of these circles. We can see that the same user belongs to multiple circles based on different identities. For instance, circle 10 and circle 18 may be circles formed by school classmates, circle 276 may be formed by work colleagues, and circle 278 may be formed by people in the same place of residence. This justifies that our method is capable of finding out the causes of the formation of distinct circles.

Table 3: The circle types of the circles to which user 2895 belongs.

Circle ID	**Circle type**
10	education;school;id
18	education;concentration;name, languages;id
276	work;employer;name
278	languages;name, location;id

5. Conclusion

In this project, we define the problem of identifying the key factors of circle formation, and further formulate it as a learning-based task. In particular, we implement logistic regression by gradient descent to address this task. Moreover, we compare our method with a naive statistical method, confirming that ours can find out important features and various circle types more accurately. Finally, we demonstrate how a user belongs to multiple types of circles, justifying the ability of our method to identify the causes of forming distinct circles.

Video link:
https://drive.google.com/file/d/1DYpFMcxuVtIJT8hcKC6fSMfjZWUfp7e2/view?usp=sharing

6. References

[1] Kaggle. 2014. Learning Social Circles in Networks. [ONLINE] Available at: https://www.kaggle.com/c/learning-social-circles/overview.

[2] Leskovec, Jure, and Julian McAuley. 2012. Learning to discover social circles in ego networks. In NeurIPS.

[3] Wikipedia. 2021. Topological sorting. [ONLINE] Available at: https://en.wikipedia.org/wiki/Topological_sorting.

[4] GeeksforGeeks. 2020. Printing pre and post visited times in DFS of a graph. [ONLINE] Available at: https://www.geeksforgeeks.org/printing-pre-and-post-visited-times-in-dfs-of-a-graph/.

[5] Hung-yi Lee, NTU. Machine Learning. [ONLINE] Available at: https://www.youtube.com/playlist?list=PLJV_el3uVTsPy9oCRY30oBPNLCo89yu49.