Data Mining Package Report

Kandati Vishnu Sai, Midhilesh E., Sanjay Seetharaman

November 2020

Abstract

With more than 2 billion monthly active users, social networking websites have dramatically changed the lives of people. With a range of features and formats, they cater to the interests of a wide range of users. People build social networks or social relationships with other people who share similar personal or career interests, activities, backgrounds or real-life connections. The availability of services on a wide range of devices such as mobile phones, laptops, and computers has made it more accessible than ever. A user can easily access other users' behaviors and is in turn influenced by them. As a result, effective social influence prediction is vital for a variety of applications such as online recommendation and advertising.

In this report, we discuss DeepInf[1], an end-to-end framework that learns users' latent feature representation for predicting social influence.

[1] Qiu, Jiezhong and Tang, Jian and Ma, Hao and Dong, Yuxiao and Wang, Kuansan and Tang, Jie. 2018. "*Deepinf: Social influence prediction with deep learning*" in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.2110–2119.

Contents

Chapter 1

Introduction

1.1 Social Network

A **Social Network** is a network that reflects the social structure of its nodes and their inter-dependency, such as friendship of people, co-authorship of researchers, and collaboration between different parties. Its structure made up of a set of social actors (such as individuals or organizations), sets of dyadic ties, and other social interactions between actors. The scale of a social network is usually very large. The dramatic development in technology has led to more and more people engage in a variety of social activities like chatting, blogging, commenting, and shopping. This has led to researchers having far more questions than answers.

A social network can be represented and treated as an abstract network of interconnected nodes, and researchers generally adopt a graph or matrix to describe the networks. However, this representation is not scalable. For large social networks, we can expect millions of nodes and edges. It is challenging

Figure 1.1: A social network diagram

to decompose or compress such huge graphs. If at all there is a provision to maintain such huge graphs, it is difficult to describe the dynamical features of social networks over such representations.

1.2 Social Influence

Social Influence is a relationship established between two entities for a specific action. In particular, one entity influences the other entity to perform an action. Usually, the first entity is called the *influencer*, the second entity is called the *influencee*. It is a function of uncertainty because the influence may not have any idea on because the influencee would perform the action. The typical examples are viral marketing, influential bloggers finding, online advertising, social healthcare, expert finding, personalized commendation, citation networks, and so on.

Influence is *asymmetric*: the fact that Alice influences Bob does not necessarily mean that Bob also influences Alice. Influence is *transitive*: if Alice influences Bob and Bob influences Carlos, this implies that Alice influences Carlos indirectly. Influence is *propagative:* information can be passed from one member to another in a social network, creating influence chains. Spreading of influence in social network is the basis of "word of mouth" propagation of information for humans

The research on social influence analysis - still in infancy - interconnects with the other features of social networks, such as influence properties, evaluation metrics of influence, collection and processing social networking big data, and selection of most influential nodes.

Analyzing social influence helps us: 1) in terms of sociology, it is helpful to understand people's social behaviors; 2) in terms of public services, it is helpful to provide a theoretical basis for public decision making and public opinion guidance; 3) in terms of political factors, it is helpful to promote national security, economic stability, economic progress, and so on. Hence, social influence analysis is significant and valuable.

There are many challenges in measuring social influence: 1) no mathematical definition; 2) uncertainty; 3) no effective way to integrate external factors; 4) characterizing the relationship.

In this report, we focus on user level influence - predicting the action status of a user given the action statuses of her near neighbors and her local structural information. For example, in Figure 1.2, consider vertex v to represent Alice. Suppose her friends Bob, Carlos, and Dave (represented by the black vertices) perform an action, say bought a product, will Alice perform the action in the future?

Researchers have proposed solutions to make such predictions, but they consider complicated hand-crafted features, which require extensive knowledge of specific domains and are usually difficult to generalize to different domains.

DeepInf is an deep learning based end-to-end approach to discover hidden and predictive signals in social influence automatically. Using techniques like

Figure 1.2: A motivating example of social influence locality prediction

network embedding, graph convolution and graph attention mechanism into a unified network, it represents both influence dynamics and network structures in a latent space.

Chapter 2

Graph Mining

Graphs are used in a variety of domains - chemical compounds, protein structures, traffic flow, program control flow, and so on. Modeling data with graphs gives enormous flexibility to the modeler for storing the elements, elements' attributes, relationship between two elements, relation between a set of elements, type of relation, and so on.

Graph Mining is essentially the problem of discovering repetitive interesting patterns/sub-graphs occurring in the input graphs. Standard data mining algorithms are based on the *flat transaction representation* (sets of items). Datasets with structures, layers, hierarchy and/or geometry often do not fit well in this flat transaction setting. A social network is one such dataset.

Some of the major tasks in graph mining are node classification, link prediction, and clustering. Traditional machine learning techniques cannot be applied directly on graphs because of the non-euclidean nature of the data structure. Analyzing graphs with machine learning has garnered a lot of interest recently, given the expressive power of graphs.

2.1 Convolutional Neural Networks

Convolutional Neural Network (CNN) is a deep learning architecture inspired by the natural visual perception mechanism of the living creatures. A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image. It has the ability to automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. The keys of *Convolutional Neural Networks (CNN)* are local connection, shared weights and the use of multi-layer structure.

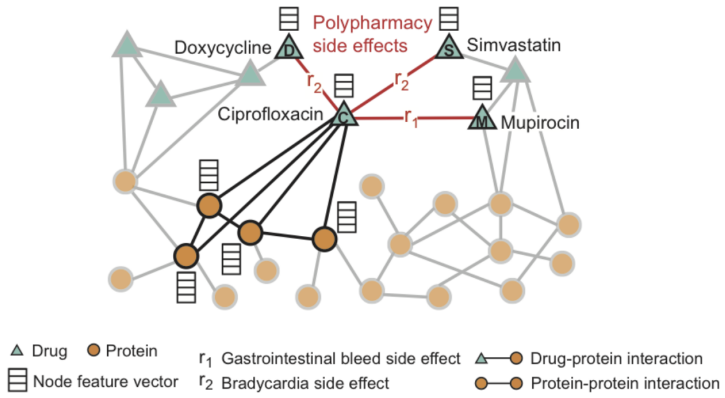


Figure 2.1: A model that predicts specific drug-drug interaction effects.

2.2 Graph Neural Networks

Graphs are locally connected structure. The shared weights reduce the computational cost compared with traditional spectral graph theory. The multi-layer structure is the key to deal with hierarchical patterns, which captures the features of various sizes. *Graph Neural Networks (GNN)* are deep learning methods that operate on graphs. They have high interpretability and acceptance. The first motivation of GNNs roots in CNNs. However, CNNs operate only on Euclidean data like images (2D) and text (1D). Also, there isn't a natural order of nodes in the graph. This requires a model to traverse all the possible orders of nodes, which is infeasible. To solve this problem, GNNs propagate on each node respectively, ignoring the input order of nodes. In other words, the GNN output is invariant for the input order of nodes.

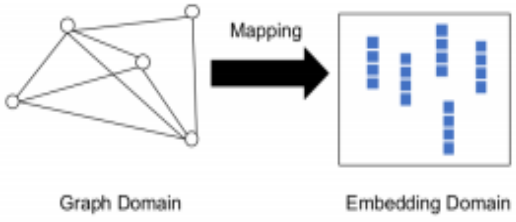
The other motivation for GNNs comes from *graph embedding*, which learns to represent graph nodes, edges or sub-graphs in low-dimensional vectors. Traditional direct embedding methods lack the ability to generalize, they cannot deal with dynamic graphs or generalize to new graphs. The target of GNN is to learn a state embedding which contains the information of neighborhood for each node. Based on CNNs and graph embedding, GNNs are proposed to collectively aggregate information from social network graphs.

Chapter 3

Network Embedding

3.1 Introduction

- Graph embedding aims to **map each node in a given graph into a low-dimensional vector representation** that typically preserves some key information of the node in the original graph.
- A node in a graph can be viewed from two domains:
  - The original graph domain, where nodes are connected via edges
  - The embedding domain, where each node is represented as a continuous vector.
- Using this embedding we can perform graph clustering, classification using the machine learning algorithms.



3.2 Algorithms

Algorithms for finding these embedding

- DeepWalk
- Node2Vec

3.3 Deepwalk

DeepWalk is a supervised learning algorithm developed to analyze graphs for classification, clustering, similarity search, and representations for statistical models.

3.3.1 Step1

Generate a set of random walks of size $\mathcal{T}$.

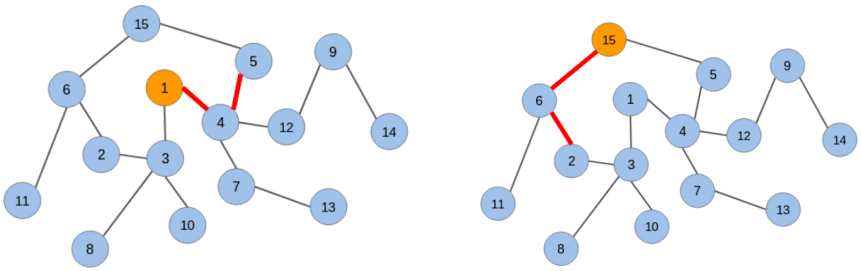
Random Walk

Let $\mathcal{G}$={ $\mathcal{V}$,$\mathcal{E}$ } be a graph. We consider a random walk starting from the node $v^{(t)}$ ϵ $\mathcal{V}$. The probability of choosing a next node is given by.

$$p(v^{(t+1)}|v^{(t)}) = \frac{1}{d(v^{(t)})}, v^{(t+1)} \epsilon \, \mathcal{N}(v^{(t)}) \tag{3.1}$$

$$p(v^{(t+1)}|v^{(t)}) = 0 \quad otherwise \tag{3.2}$$

Example of random walk with length 2



(a) random walk 1 (b) random walk 2

3.3.2 Step2

Considering the set of random walk as set of vocabulary for which **Skip gram** model can be used to find the node embedding.

Skip gram model

Skim gram is algorithm in language modeling tries to preserve the information of the sentences by capturing the **co-occurrence relations between words in these sentences**.

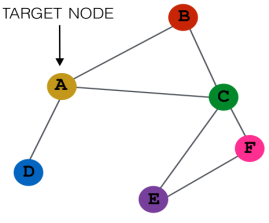
Chapter 4

Graph Neural Network's

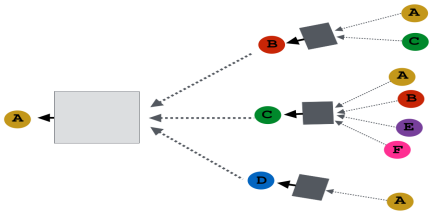
4.1 Introduction

- GNN defines a class of functions for relational reasoning over graph-structured representation.
- It uses **neighbourhood aggregation** and **graph pooling** to solve the graph related problems
- Neighbour aggregation contains two important steps
  - **Aggregation** - aggregating information from adjacent nodes, edges
  - **Updation** - updating the target node with the aggregated neighbourhood information.
  - **Note**: Aggregation should be **permutation invariant operation** eg., Sum, Mean, Max

4.2 GNN Pictorial Representation



(a) Input Graph



(b) Neighbourhood aggregation (2-layer model)

4.3 Neighbourhood aggregation

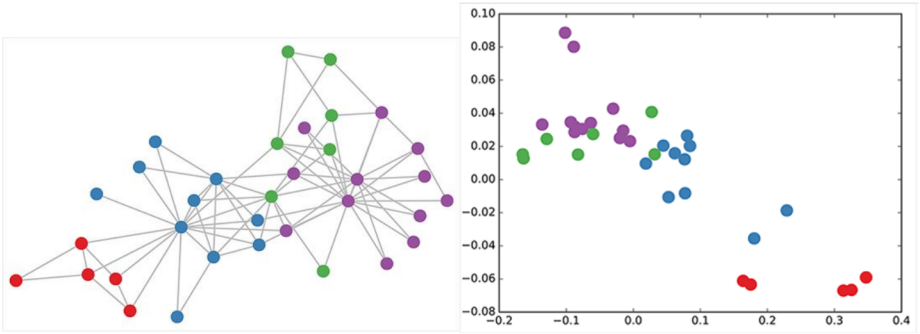
Let's assume that we have only the node features and it is represented h_v for the node v. Let N(v) denotes the neighbourhood of node v.

$$h_v^k = \sigma(W^k \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|} + B^k h_v^{k-1}) \tag{4.1}$$

The above equation represents the k-th layer of GNN with **mean aggregation** and **Neural Network updation**

4.4 GNN Interpretation

- Zachary Karate Club social network, where nodes are connected if the corresponding individuals are friends.
- The nodes are colored according to the different communities that exist in the network.



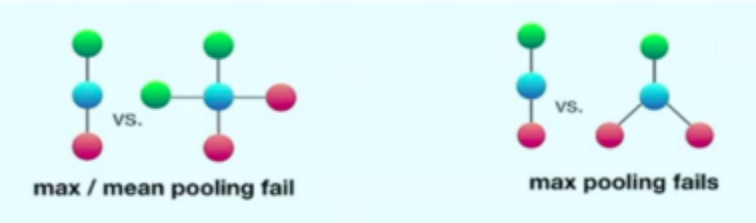
(a) karate club dataset (b) Node embedding of GNN

4.5 Limitation of Neighbourhood aggregation

The blue node represents the target node and it takes information from the neighbour nodes. The green and red nodes are the neighbour nodes.

In mean pooling, the information gain from neighbour nodes will remain same irrespective of the dimensions. From the figure, it is shown that single dimension or two dimensions make no difference in information acquired as the no.of nodes are proportionally equal.

In max pooling, the information gain remains same as the max node will be populated in the next dimension which provides the same information. So, the max or mean pooling fails.



Chapter 5

Graph Convolutional Neural Network

5.1 Definition

Graph Convolutional Network (GCN) is a semi-supervised learning algorithm for graph-structured data. This is refered convolutional, because filter parameters are typically shared over all locations in the graph.

GCN model goal is to learn a function of signals/features on a graph $G = (V, E)$ which takes input:

- A feature description x_i for every node i,summarized in a $N \times D$.
- A representative description of the graph structure in matrix form; typically in the form of an adjacency matrix A.

It produces node level output Z (an $N \times F$), where F is the number of output features per-node.

5.2 GCN mathematical formulation

A simple neural l-th layer network as per above definition can be formulated as

$$f(H^l, A) = \sigma(AH^lW^l) \tag{5.1}$$

The main limitation on the above formulation is A is not normalized therefore the multiplication with A will completely change the scale of the feature vectors. The adjacency matrix can be normalized (making row sum equal to 1) by multiplying inverse of the diagonal matrix with has degree value of each node in the diagonal(D)(i.e $D^{-1}A$)

The mathematical fomulation of GCN as follows:

$$f(H^l, A) = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{\frac{1}{2}}H^l W^l) \tag{5.2}$$

where $\hat{A} = A + I$ is the adjacency matirx after adding self loop and I is the identity matrix, $\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{\frac{1}{2}}$ is the symmetric normalization for better convergence rather than normal normalization and σ is the non-linear activation functon.

5.3 GCN pictorial representation

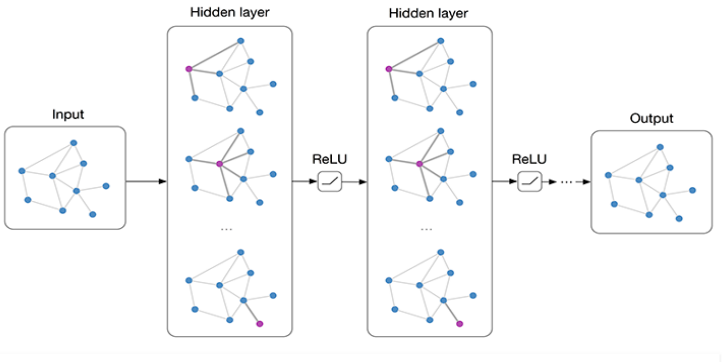


Figure 5.1: multi-layer GCN network

The above figure 5.1 represents the pictorial representation of multi-layer GCN with one hidden layer. At each layer the model learns the node features with respect to grap structure and the node's inherient features.

Chapter 6

DeepInf: Social influence prediction

6.1 Introduction

Deepinf focus on the prediction of user-level social influence. It aims to **predict the action status of a user given the action**. It is a deep learning based framework to represent both influence dynamics and network structures into a latent space and tries to minimize the negative likelihood that was defined in the section 1.

6.2 Problem Formulation

Let $G = (V, E)$ be a static social network, where V denotes the set of users and $E \subseteq V \times V$ denotes the set of undirected relationships. For a user v, its **r-neighbors** are defined as $\Gamma_v^r = \{u : d(u, v) \leq r\}$ where $d(u, v)$ is the number of hops (shortest path distance) between u and v in the network G. The **r-ego network** of user v is the subnetwork induced by Γ_v^r, denoted by G_v^r.

Users in social networks perform **social actions**, such as retweet. At each timestamp t, we observe a binary action status of user u, $s_u^t \in \{0, 1\}$, where $s_u^t = 1$ indicates user u has performed this action before or on the timestamp t, and $s_u^t = 0$ indicates that the user has not performed this action yet. Such an action log can be available from many social networks.

Given the above definitions, **social influence locality** can be stated as: users' social decisions and actions are influenced only by their near neighbors within the network, while external sources are assumed to be not present.

Problem 1. Social Influence Locality. Social influence locality models the probability of v's action status conditioned on his r-ego network G_v^r and the action states of her r-neighbors. More formally, given G_v^r and $S_v^t = \{u \in \Gamma_v^r$ $\{v\{\}$, social influence locality aims to quantify the activation probability of v

after a given time interval Δt:

$$P\big(s_v^{t+\Delta t}|G_v^r, S_v^t\big)$$

We formulate social influence prediction as a binary graph classification problem which can be solved by minimizing the following negative log likelihood objective w.r.t. model parameters Θ:

$$L(\Theta) = -\sum_{i=1}^{N} \log\left(P_\Theta\big(s_{v_i}^{t_i+\Delta t}|G_{v_i}^r, S_{v_i}^{t_i}\big)\right)$$

6.3 Model framework

6.3.1 Sampling Near Neighbour

Give a user v, a r-ego network G_v^r is extracted using breadth-first search (BFS) starting from user v. However, G_v^r may have different size due to the small world property in the social network. Since most deep learning models expects fixed size data, the graph G_v^r can be sampled to fixed size.

For sampling a fixed size graph **random walk the restart** (RWR) was used. RWR algorithm is defined as following steps.

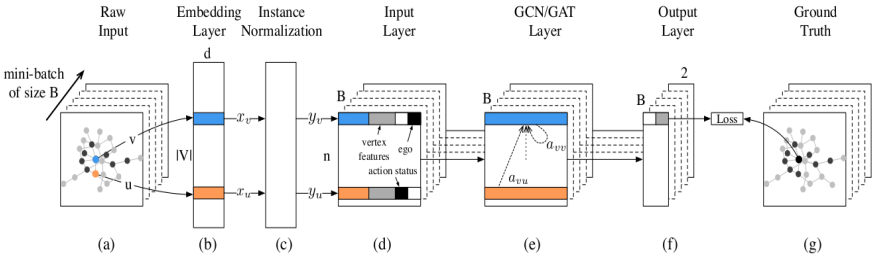
- Start random walks from either the ego user v or one of her active neighbors randomly.
- The random walk iteratively travels to its neighborhood with the probability that is proportional to the weight of each edge.
- At each step, the walk is assigned a probability to return to the starting node, that is, either the ego user v or one of v's active neighbors.
- Run the algorithm until a fixed number of vertices denoted by $\hat{\Gamma_v^r}$ with $|\hat{\Gamma_v^r}|$=n.

After running this algorithm, a sub-graph $\hat{G_v^r}$ and denote $\hat{S_v^t}$= { $s_u^t : u \in \hat{\Gamma_v^r}$ } be the action statuses of v's sampled neighbours. Therefore we re-define the optimization objective in section 1 as:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log(P_\theta(s_{v_i}^{t_i+\Delta t}|\hat{\mathcal{G}_{v_i}^t}, s_{v_i}^{t_i})) \tag{6.1}$$

6.3.2 Neural Network

With the retreived $\hat{G_{v_i}^t}$ and $\hat{S_v^t}$ for each user, an effective neural network model to incorporate both the structural properties in $\hat{G_{v_i}^t}$ and action statuses in $\hat{S_v^t}$.



Deepinf neural network model consist of network embedding layer, instance normalization layer, input layer and GCN layer as described in the above diagram.

Embedded layer

Network embedding technique encode network structural properties into low dimensional matrix $\mathbf{X} \in \mathbf{R}^{\mathcal{D} \times |V|}$. Deepinf uses Deepwalk algorithm for mapping each users into $\mathbf{R}^{\mathcal{D}}$ space.

Instance Normalization

Instance normalization can remove instance-specific mean and variance, which encourages the downstream model to focus on users' relative positions in latent embedding space rather than their absolute positions. It also prevents overfitting.

Let x_u be the low dimension representation for the user $u \in \hat{\Gamma}_v^r$, the instance normalized vector y_u is obtained by

$$y_{ud} = \frac{x_{ud} - \mu_{ud}}{\sqrt{\sigma_d^2 + \epsilon}} \tag{6.2}$$

for each embedding dimension d = 1...$\mathcal{D}$, where

$$\mu_d = \frac{1}{n} \sum_{u \in \hat{\Gamma}_v^r} x_{ud}, \;\; \sigma_d^2 = \frac{1}{n} \sum_{u \in \hat{\Gamma}_v^r} (x_{ud} - \mu_{ud})^2 \tag{6.3}$$

Input Layer

Input later constructs feature vector for each user. Along with the output from the instance Normalization, input layer adds two binary variable. The first variabe indicates user's action status and second variable indicates whether the user is the ego user.

GCN

Graph Convolutional Network (GCN) was discussed in detail in the previous chapter. Given the node features matrix $\mathcal{H} \in \mathbf{R}^{n \times F}$, for n nodes in the subgraph, GCN tries to extract neighbourhood information for all the nodes in the subgraph and projects into lower dimensional space.

Output layer and loss function

The output layer produces a two dimension representation for each users and comparing the representation for the ego user with the ground truth value the negative log-likelihood (Eq 6.1) is minimized.

6.4 Experimentation

6.4.1 Dataset

Digg

- Digg is a **news aggregator which allows people to vote web content, a.k.a, story, up or down**.
- The dataset contains data about stories promoted to Digg's front page over a period of a month in 2009.
- For each story, it contains the list of all Digg users who have voted for the story up to the time of data collection and the time stamp of each vote.

Data preparation

- As dataset is imbalanced (negative instances are more than positive instances), the data is sampled to achieve a balanced distribution.
- Each subgraph for the user v, has fixed neighbors (50).

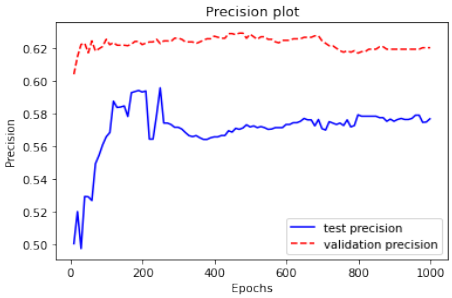
6.4.2 Evaluation metrics

The model was trained for 1000 epochs and prediction performance like AUC, Precision, Recall, F1-Score were calculated.

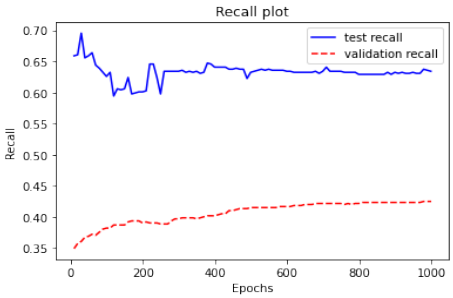
Data	Model	AUC	Precision	Recall	F1
Digg	DeepInf+GCN (Our's)	0.82	0.58	0.634	0.604
Digg	DeepInf+GCN (Author's)	0.84	0.587	0.676	0.6288

Precision Recall plots

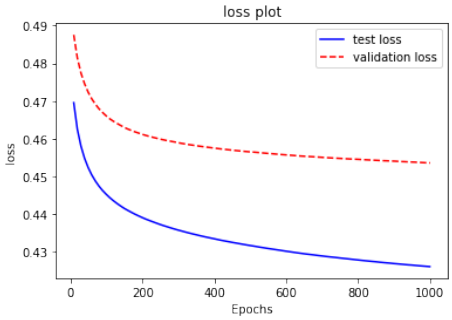
Loss f1-score plots



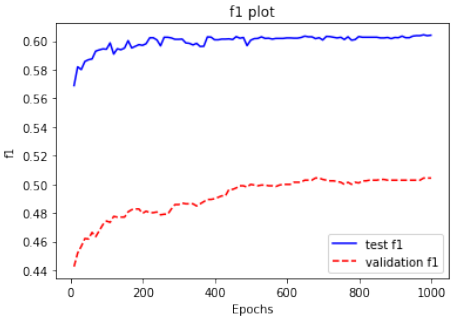
(a) Precision plot



(b) Recall plot



(a) Loss plot



(b) F1-score plot