Interdisciplinary studies of Complex Network and Machine Learning and its applications

by

Shaojun Luo

A dissertation submitted to the Graduate Faculty in Physics in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2018

Interdisciplinary studies of Complex Network and Machine Learning and its applications

by

Shaojun Luo

This manuscript has been read and accepted by the Graduate Faculty in Physics in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

	Professor Hernan Makse
Date	Chair of Examining Committee
	Professor Igor Kuskovsky
Date	Executive Officer

Dr. Gino Del Ferraro

Dist. Prof. Robert Haralick

Dr. Flaviano Morone

Prof. Lucas Parra

Supervisory Committee

The City University of New York

Abstract

Interdisciplinary studies of Complex Network and Machine Learning and its applications

by

Shaojun Luo

Adviser: Professor Hernan Makse

This dissertation consists of Five chapters...

Chapter 1 Driven by the unprecedented rapid growth of data science, Network Models are widely used in statistical inference model due to its advantages of direct representation of pairwise interaction, convenience in inference based on variable dependency, and numerous degree of freedom to develop complex models. However, due to the complex nature of network, many restrictions are applied to network models in order to make the models practically solvable. We will discuss the network-based statistical models and give two example of its application in the following chapter.

Chapter 2 The network-based statistical methods are divided into two types: network structure inference and variable inference. For network structure inference, we introduce correlation matrix, graphical Lasso, network clustering and identify the influencer in the network. For variable inference, we also introduce from Bayesian network, to Random Markov Field and Ising Model, Boltzmann and Restricted Boltzmann machine and the algorithm of Belief Propagation. Last but not the least, we introduce the most widely used neural network family and its two main types: Convolutional Neural Network and Recurrent Neural Network.

Chapter 3 A concrete example of applying network structure inference algorithm to find the correlation between network metrics and socio-economic stats are introduced in this chapter. A mobile network with 10^8 nodes were constructed from mobile records to build the social networks by filtering the abnormal phone lines with a semi-supervised learning. Collect Influence (CI) is used as the proxy of network influence. A novel correlation ($R^2 = 0.95$) is achieved by investigating the correlation between aggregated population which is based on both age and network metrics quantile. The result is validate by a marketing campaign.

Chapter 4 In this chapter, we provide an example of combining large scale neural networks to build a deep learning workflow to predict the pathology result of breast tumor based on MRI images. The workflow consist of three agents: Feature Extraction Agent which is a deep convolutional neural network transferred from inception v3. Image Selection Agent is a bi-directed recurrent neural network which evaluate the score of risk for each slice window and a Pathology Prediction Agent is to predict the pathology result based on the slices windows based on selection agent. The workflow is trained by reinforcement learning in order to automatically detect the location of tumor. Although the result indicates the workflow is able to capture the evidence of malignancy, the workflow still needs to be improve to increase stability.

Chapter 5 At the end of dissertation, we discuss about the underlying problems of incomprehensibility, low efficiency and limitation of application for network-based inference model. Currently major progress has been made by Hinton with a new concept of a capsule network. This concept has largely enriched the insights of general network based statistical inference models.

Acknowledgments

I would like to express my deepest appreciation to my advisor, Professor Hernán Makse for the patient guidance, inspiration, encouragement, advice and financial support he has provided throughout my time as his student. It is lucky to become a student of such passionate, genius and responsible supervisor who has collective knowledge and insight of research and cares about student. In addition, I would like to give my special thank to my Committee member, Distinguished Professor Haralick, who reviewed and corrected this work with his incomparable expertise. I would also like to thank to the committee member, Professor Lucas Parra for his valuable discussion and professional instruction. And my thanks Dr. Gino Del Ferraro and Dr. Flaviano Morone for serving as committee member and the contribution to the work.

I must express my gratitude to Grandata Inc. and Memorial Sloan Kettering Cancer Center for the data support, and the Research Foundation and Physics Department of City University of New York for their financial support. The work would not have been possible without them. I also want to thank Qionge Li, Kate Burleson-Lesse, Francesca Lucini and Zhuo Yin for their discussions for the work.

Last but not the Least, I would like to express my deeply thank to my family whose love and support are with me in whatever I pursue. I addition, I would like to give my special thank to my beloved Xiaoyu, whose accompany, support and guidance help me to overcome the days of long march before graduation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Driven by the unprecedented rapid growth of demand in big data analysis, data science has become one of the merging research area nowadays [2]. Like all the "new sciences" like materials science, data science is an interdisciplinary field of science which study across multiple aspects of research area. By definition, it is an ensemble of scientific methods, processes, algorithms and systems to extract knowledge or insights from data in various forms, either structured or unstructured [3]. The research subject and application scenes of the data science varies from fundamental physics and chemistry [4, 5], biological and medical science [6, 7], operation research and business [8], social and political science [9] or even the daily life of people [10]. Although the application of data science techniques range widely, the methodology behind the modeling is almost invariant.

One of the key components for data science methodology is statistical inference. The main purpose for statistical inference is to construct the insight of variables from a given amount of data. This insight includes the dependencies between different variables, the significance of effects, the detection of hidden variables and causality discovery [2]. One of the commonly known concepts in statistical inference is machine learning. Machine learning is a field of computer science that uses statistical techniques to give computer systems the

ability to progressively "learn" with data without being explicitly designed for the specific task [11]. Most machine learning techniques are based on a "statistic machine" which means they are learning through the statistical outcomes from the data. Usually such machines mimic the learning process of human by applying a common structure of statistical inferring workflows. The design of such structure is the basis for designing machine learning models.

1.1 Introduction of Network Based Inference Model

Many physics concepts, especially those in thermostat, have been implemented in modeling the learning structure [12]. For example, one of the famous application is the use of first principles on energy and entropy in inferring the distribution of variables [13]. Another common interdisciplinary application of physics concept is the use network structures. The importance of introducing network are obvious:

1. **The network is the most direct way in characterizing the pair-wise interactions between samples/variables.** The sketch of the networks is convenient for describing the conditional dependency between different features (for example, the Bayesian network in the next chapter) which helps us to construct a better understanding for the data.

2. **The network is efficient in expressing the complex dependencies between high dimensional variables.** Many machine learning techniques like regression [14], SVM [15] and KNN [16] works very well in low dimensional cases. However, for high dimensional but low-rank problems, the interpretation of the method above are difficult due to the sparseness in the variable space [17]. The network representation overcomes the sparseness by projecting the variables space into a compact and dimensionless model. In the network representation, variables are compressed into nodes inside a network. Fig. 1.1 shows an simple example for the procedure.

3. **The elasticity of network structures makes it possible to develop deep and complex models** Unlike data structure based learning models like decision trees, network based models have more flexibility in developing the complex structures including multiple dependencies and loops (Fig. 1.2). Such elasticity is the structural basis of deep learning designs.

Due to the advantages above, network-based models have become the most popular infrastructures for deep learning tasks. It is widely used in image recognition [12, 18], natural language processing [19] and even the development of artificial intelligence [20].

Nevertheless, the network has its disadvantages. The main disadvantage for the network based model is the difficulties in training. Due to the complexity of dependencies between variables (nodes), the network is a non-linear system which in many cases the convexity and convergence of loss function can not be guaranteed [21]. For example, for the simplest Bayesian Network model, it costs $O(n^4)$ of time complexity in training which makes it impossible to apply for a large dataset. When it comes to more complicated models like Boltzman Machine or Random Markov Field, restrictions on the network structures are applied (usually is locally tree like) otherwise the training will likely to be fallen into local minimums [22].

By carefully designing the structure, it is possible to train a deep neural network by stochastic gradient descent [23]. However, there are considerable practical problems during the training including the gradient vanishing [24], slow descending rate around a saddle point [25] and unpredicted local minimums [26]. Although various techniques have been developed to overcome the problems [27], the process of training still needs to be carefully fine-tuned due to its complexity.

1.2 Purpose and Contents

In this dissertation, the author will review the concepts of network based statistical inferring models. Despite the models being developed independently in different areas (computer science, physics, biology, etc.), the author will try to unify all these models under the frame of thermostat theory in physics to obtain a better understanding of network based inference models. Two examples are also given to demonstrate how the network modeling and solve the practical problems in two different ways.

There are five chapters in the dissertation:

Chap 1 Chapter 1 gives the introduction of the backgrounds knowledge of the dissertation.

Chap 2 Chapter 2 gives brief reviews of different statistical inference methods based on network models.

Chap 3 Chapter 3 gives an example on how to construct and implement statistical inferring model on networks.

Chap 4 Chapter 4 gives an example on how to design and implement network based deep learning framework.

Chap 5 Chapter 5 gives the conclusion and further discussions based on the works done in the dissertation.

For more detailed information, please refer the specific chapter and their abstracts.

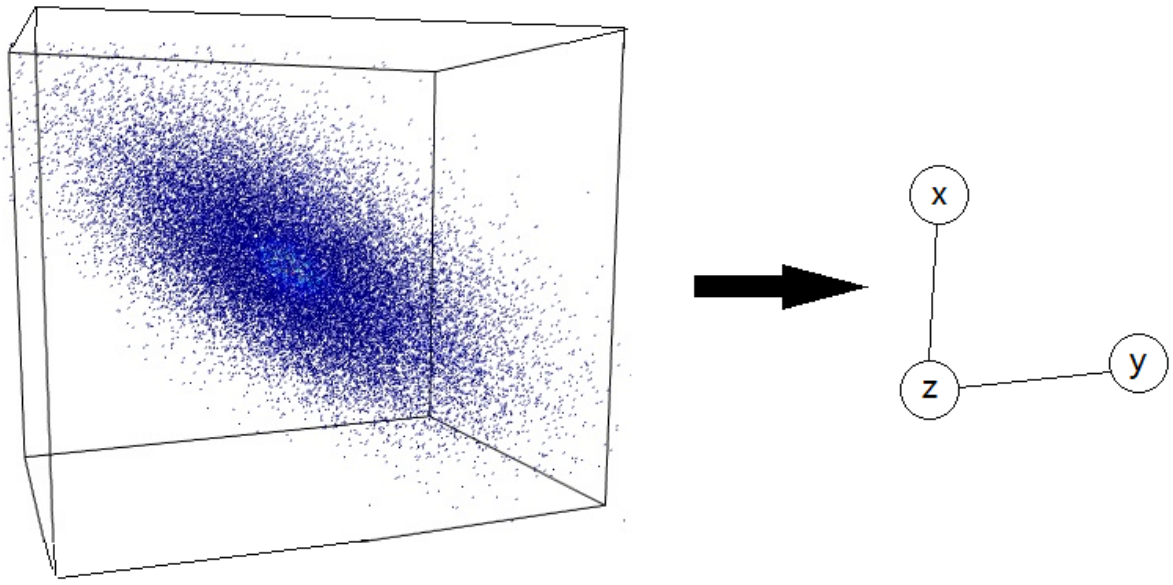

Figure 1.1: Network Representation of a 3D variables space

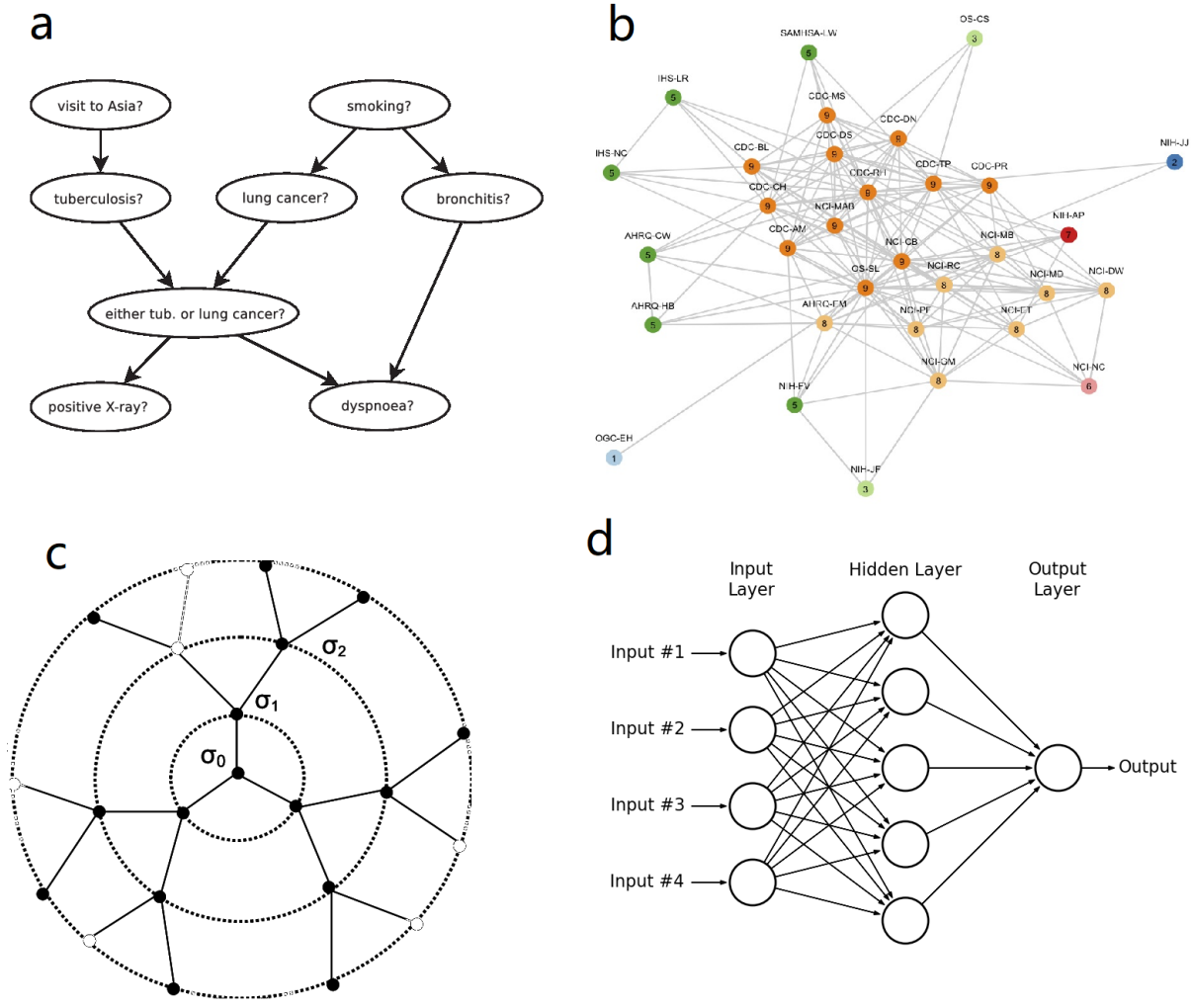

Figure 1.2: Types of Network Inference Models
a) Directed (Bayesian Network [28]) **b)** Undirected (Random Markov Field [29]) **c)** Tree-like (Bethe Lattice [30]) **d)** Layer-based (Neural Network [31])

Chapter 2

Network Based Statistical Inference Models

In this chapter, we will briefly introduce the network models are used in statistical inference. Such application is not limited to machine learning techniques. Generally, network based model is an ensemble of nodes N and edges E: $G(N, E)$. The nodes are associate with variables, and the edges represent the connections and coupling rules between neighboring variables. The structure can be both static and dynamic. However, in most cases we assume the structure of networks remains static.

There are two types of statistical inference for a network model. 1. Construct and infer the structure of network itself from data. 2. Infer the value of variable nodes/edge for a given network structure. Both types of network inferences problems will be described in this chapter.

2.1 Network Structure Inferring

A network can be easily constructed by setting up the link between two nodes whether the link is physically exist or not. For a concrete network like the contact network of packed balls or social networks, we just need to infer the intensity (weights) of the edges according to a specific task. For an abstract network which represented the correlations between two variables, the construction methods may vary according to how the researchers defined "correlation".

Generally, if the network is constructed from empirical correlation, it is necessary to use statistical methods to infer the real structures. Such inference methods also vary based on how the researchers define "correlation"

2.1.1 Correlation matrix and percolation

In this subsection, we introduce the most common methods in constructing the network: the Correlation Matrix.

Assume we have n variables with unknown dependencies, correlation matrix C_{ij} is an $n \times n$ matrix representing the pairwise correlation between two variables x_i, x_j:

$$C_{ij} = f(x_i, x_j) \tag{2.1}$$

where f is the correlation function such as Pearson Correlation [32], Spearman Correlation [33] or other shared functions (like the co-occurrence probability of events, etc). The matrix is symmetric and positive semi-definite which results in the networks being undirected. In practice, we usually assume C_{ij} is sparse since most of the variables are independent.

In addition, if we get the m observation for the paired variables $(\mathbf{x_i}, \mathbf{x_j})$ one can infer the correlation matrix by

$$\hat{C}_{ij} = \hat{f}(\mathbf{x_i}, \mathbf{x_j}) \tag{2.2}$$

where f is the estimator of correlation function. If the estimator is unbiased, according to the law of large numbers, the empirical correlation matrix will converge to the correlation matrix. $\hat{C}_{ij} \overset{m\to\inf}{\to} C_{ij}$. Thus the it is very important to choose the estimators and ensure an adequate amount of observations ($m >> n$).

However, in most cases, the number of observations are not sufficient to get a good estimation of correlation matrix. A common solution for the problem is to turn the network into a parametric model controlled by a threshold p and simply cut the values with the threshold:

$$\hat{C}_{ij} = \begin{cases} \hat{f}(\mathbf{x_i}, \mathbf{x_j}) & \hat{f}(\mathbf{x_i}, \mathbf{x_j}) \geq p \\ 0 & \hat{f}(\mathbf{x_i}, \mathbf{x_j}) < p \end{cases} \tag{2.3}$$

Different thresholds will result in different structures of network. If we plot the size of Giant Component (largest connected components) vs. the threshold p, we get a curve indicating the sparseness of the network. Fig. 2.1 is an example for the percolation process. We discovered the decrement of giant component size when the threshold increases. When the threshold reaches a critical point p_c, the network will totally collapsed. Such phenomena is called "percolation" which describes the phase transition in the network structures[34]

2.1.2 Graphical Lasso

Consider a variable space $\overrightarrow{x} = (x_1, x_2, ..., x_n)$ which follows a multivariate normal distribution:

$$N(\overrightarrow{x} : \overrightarrow{\mu}, \mathbf{C}) = \frac{1}{\sqrt{(2\pi)^n \mathbf{C}^{-1}}} \exp[-\frac{1}{2}(\overrightarrow{x} - \overrightarrow{\mu})^T \mathbf{C}^{-1} (\overrightarrow{x} - \overrightarrow{\mu})] \tag{2.4}$$

where $\overrightarrow{\mu} = (\mu_1, \mu_2, ..., \mu_n)$ is the expectation value of each variable and $\mathbf{C}$ is the covariance matrix. In practical use, we usually choose the inverse of the covariant matrix $\mathbf{J} = \mathbf{C}^{-1}$ as the representation for the relations because it is easy to calculate the z-statics. One can easily prove that the location of non-zero element of $\mathbf{J}$ is the same as $\mathbf{C}$. Thus the structure of networks are the same on both representations.

The application of Graphical Lasso is used for inferring the inverse covariance matrix from the empirical covariance matrix $\hat{\mathbf{C}}$. Assume we have m observations, the unbiased estimator of covariance matrix is:

$$\hat{C}_{ij} = \frac{1}{m-1} \sum_{k=1}^{m} (x_{ki} - \mu_i)(x_{kj} - \mu_j) \tag{2.5}$$

where the $\overrightarrow{x_k} = (x_{k1}, x_{k2}, ..., x_{kn})$ is vector for the k-th observations. Due to the lack of samples, empirical covariance matrix are usually not sparse. Thus the direct inversion of the empirical covariance matrix will result in a bad estimation. In order to reduce the non-zero elements of the inferred matrix, we introduce a 1-norm penalty term with parameter λ under the maximum log-likelihood estimation [35]:

$$\mathbf{J} = \underset{\mathbf{J} \succ 0}{\operatorname{argmax}} \log\det(\mathbf{J}) - \mathbf{tr}(\mathbf{J}\hat{\mathbf{C}}) - \lambda ||\mathbf{J}||_1 \tag{2.6}$$

The first two terms will reach maximum when $\mathbf{J} = \hat{\mathbf{C}}^{-1}$. However, because of the existence of the regularization term $\lambda||\mathbf{J}||_1$, the result matrix will be more sparse than the direct inverse. In this case, the elements of $\mathbf{J}$ can be obtained by quadratic gradient descend over the functions above [36]. Fig. 2.2 shows the inference result of for a sample matrix. Same as the threshold p, we can also choose a good estimation of original covariance matrix by choosing the correct λ.

2.1.3 Network clustering

Once the structure of the network is determined, we are able to renormalize the network by aggregating the variables clusters with high dependencies. To determine the "clusters" in network is an open question because there is no clear definition of clusters in network. However, heuristically, we can aggregated the nodes by unsupervised learning approaches just like the k-means algorithm [37].

In this work, we introduce a similar fast unsupervised approach on positive weighted network community detection which is developed by Blondel *et al.* [38]. The algorithm defines a function called modularity [38, 39]:

$$Q_m = \frac{1}{W}\sum_{i,j}[W_{ij} - \frac{W_i W_j}{W}]\delta(c_i, c_j), \tag{2.7}$$

where W_{ij} is the weight of link i, j and c_i is the community label of node i. $W_i = \sum_{j\in\partial i} W_{i,j}$ where ∂i denote the nodes adjacent to i and $W = \sum_{i,j} W_{i,j}$. The clustering process is to maximize the modularity function by continuously coloring the nodes greedily. The algorithm is presented as follows:

1. At the beginning, each node in the network is assigned to its own community.

2. For each node i, calculate the change in modularity for removing i from its own community and joining it into the community of each neighbor j of i. The change of modularity is ΔQ

3. If $\Delta Q > 0$, then relabel the node i to the same community of node j. Otherwise keep the current community labels.

4. After update the community label sequencely. Repeat 2,3 until there is no updates on the community labels.

The global maximization of modularity was achieved by iteratively updating the modularity. However, we notice that, the final result may not be identical if we choose a different order of nodes to update the modularity function. Fortunately, in most cases, the algorithm can capture the most important clustering structures despite the order of updating. Also, during the different epochs of the iterations, we could obtain communities of different levels. The fast and flexibility of the algorithm makes it the most used network clustering algorithm.

2.1.4 Influence of nodes

Instead of aggregate variables into a connected structures, sometimes we need to evaluate the importance of variable/nodes according to the network structure. The definition for important variables/nodes also varies under different tasks. Generally, we use the "centrality" to evaluate the influence of network nodes. Higher centrality means a node is generally easier to affect / be affected by other nodes which may imply the node plays an important role in generating/passing the effect of other variables.

Many metrics are used in network theory in inferring the centrality of nodes inside a network. The metrics can either be local, regional or global. Most metrics are developed by heuristic approaches but some of them are through mathematic processes for a specific problem. Here we introduce some common measurement of centralities.

1. **Degree** Degree is the number of edges incident to the node, with loops counted twice. It is the simplest local metrics in evaluating the centrality. Sometimes the degree may be the weighted sum of edges $k_i = \sum_{j \in \partial i} W_{ij}$.

   In a directed graph, we can also apply the concept of in-degree and out-degree where in-degree is the number of edges pointed to the node while the out-degree is the number of edges pointed from the node.

2. **k-shell** k-core and k-shell index k_{s} [40] capture the centrality of a node in the global

network by the method of k-shell decomposition. In this method, nodes are removed iteratively if their degree $k_i < k$ until all the remaining nodes have degree equal to or greater than k. These nodes remain in the k-core of index k. The largest k-core a node can hold is the k-shell index k_s, which means the node is in the 'shell' of the k'th core but outside the $k+1$'th core. The k-shell or k-core number is a global metric. It has been proven efficient in identifying single influencers through the SIR model [40]. The k-shell index requires the overall information of the network. It is a quantity that does not allow one to classify the nodes with high resolution: there usually exist a few k-shells in the whole system, each containing many of the nodes in the network. Fig. 2.4a is a schematic example of a k-shell in a network.

3. **Betweenness Centrality** betweenness centrality [41] is a global metrics in evaluating the influence of network. For every pair of nodes in a connected network, there exists at least one shortest path between the nodes such that either the number of edges that the path passes through (for unweighted graphs) or the sum of the weights of the edges (for weighted graphs) is minimized. The betweenness centrality for each nodes is the number of these shortest paths that pass through the node.

   According to the definition, betweenness centrality capturing both the nodes locate at the "center" of the network but also the critical nodes connected multiple clusters 2.3. However, the computational time for Betweenness Centrality is costive ($O(n^3)$) which makes it not suitable for a large scale network.

4. **PageRank** PageRank [42] is as eigenvalue centrality metric used to evaluate the probability that information or knowledge will likely visit a node through a random walk. PageRank is calculated through an iterative algorithm in which nodes collect PageRank values from their neighbors in every iteration. For simplicity, each node is initially assigned a value of $\mathrm{PR}(i) = 1$. During each iteration, node i collects a PageRank value

through the link pointed from its neighbor j ($j \to i$) as the PageRank PR(j) of an adjacent node divided by its outbound degree k^j_{out}. Namely,

$$\text{PR(i)} = (1-d) + \sum_{j \in (\partial i \to i)} \frac{\text{PR}(j)}{k^j_{\text{out}}}. \tag{2.8}$$

Here $\partial i \to i$ is the set of points which have outbound links to i, and d is a damping factor which we choose as 0.7 in our work. When a converging threshold (10^{-4}) is reached, the iteration stops and outputs the final result of PageRank.

Although PageRank was originally proposed for ranking websites, it has also been applied in social network analysis. Given the assumption that senders of messages or makers of phone calls are likely to be the ones providing the information being communicated, PageRank is a good metric to evaluate the likelihood that an individual captures the information spreading in the network. Similarly to k-shell, PageRank requires the global information of the whole network. However, it is easy to update when the network changes.

5. **Collective Influence** Collective Influence (CI) is an algorithm to identify the most influential nodes via optimal percolation [43]. Rather than the above heuristic metrics, Collective Influence is introduced by a theoretical approximation of the solution to a problem of influence maximization in locally tree-like social networks [44]. CI minimizes the largest eigenvalue of a modified non-backtracking matrix of the network in order to find the minimal set of nodes to disintegrate the network. It has been shown that this process maximizes the spread of information via a threshold model of spreading and also provides the most important nodes for the integrity of the network (optimal percolation). Each node is associated with a CI value, and those with the top CI values are the most influential nodes in the network. The definition of CI is given by:

$$\mathrm{CI}(i) = (k_i - 1) \sum_{j \in \partial \mathrm{Ball}(i,\ell)} (k_j - 1), \tag{2.9}$$

where the Ball(i, ℓ) is all the nodes with distance ℓ. We should note that the mobile communications network is a typical small world network (average path length $< \ell > \sim$ 8.9), and the radius ℓ of the ball is limited by the network diameter.

6. **Network Diversity** Network Diversity [45] is a local metric defined on a weighted network. It equals to the Shannon entropy of edges normalized by the degree of node i:

$$D_i = \frac{-\sum_{j \in \partial i} p_{ij} \log p_{ij}}{\log k_i} \tag{2.10}$$

Where the k_i is the degree of node i and $p_{ij} = \frac{W_{ij}}{\sum_{j \in \partial i} W_{ij}}$ is the normalized weights of the link (i, j). Network diversity measures how evenly the node interact with the neighbors.

The definition can be also extended to the networks where the definition of clusters are clear (Like geo-network). Since the clusters can be obtained by fast clustering methods we mentioned in 2.1.3, it is possible to apply the concept to general networks. The extended diversity is defined by follows:

$$D'_i = \frac{-\sum_{j \in \partial i} p_{c_i c_j} \log p_{c_i c_j}}{\log k_i} \tag{2.11}$$

where $p_{c_i c_j} = \frac{\sum (j \in \partial i | j \in c_j)}{k}$ is the fraction of the links connected from i to the nodes inside the cluster c_j. The community diversity capture part of the betweenness centrality feature while keeping a very efficient computational time.

2.2 Network Based Machine Learning Method

In this section, we introduces some commonly used network models in inferring the values of variables when the structure is fixed. The basic idea in solving such kind of problem is to propagate the change of variables through the coupling rules of edges until converge. For different network models under different prior assumptions, different coupling rules are applied. Generally, the models are not practical either due to the complexity of its structure or the cost of computation resources. However, they are a perfect demonstration of how the network structures affect the value of variables.

2.2.1 Bayesian Network

Bayesian Network [28] are constructed based on the coupling rule of Bayesian assumptions. Given a random variable space $\mathbf{x}$ without any prior knowledge about dependencies. The joint distribution of the variables is:

$$\mathrm{P}(X_1 = x_1, \ldots, X_n = x_n) = \prod_{v=1}^{n} \mathrm{P}\left(X_v = x_v \mid X_{v+1} = x_{v+1}, \ldots, X_n = x_n\right) \tag{2.12}$$

There are N products in the equation above which will result in a very complicated probability space. The inference will encounter problems where there are not adequate samples to infer all the terms of product.

If we already know the dependency relations, we are able to construct a unlooped directed network which describes the dependencies of the variables (See Fig. 2.5). The nodes without connections of the links are independent based on the assumption. Thus the joint distribution can be reduced to:

$$\mathrm{P}(X_1 = x_1, \ldots, X_n = x_n) = \prod_{v=1}^{n} \mathrm{P}(X_v = x_v \mid X_{j \in \bar{\partial} v} = x_j) \tag{2.13}$$

The conditional probability for inference can also be reduced to:

$$\mathrm{P}(X_v = x_v \mid X_{i \neq v} = x_i) = P(X_v = x_v \mid X_{j \in \bar{\partial} v} = x_j) \tag{2.14}$$

where $\bar{\partial} v$ means the parent node of node v. By propagating the probability distribution from the top of Bayesian network. One can infer the distribution of all other variables depend on these variables. The representation of Bayesian network is simple and powerful. Nevertheless, it requires a carefully selected network which reflects the correct correlation of variables.

Particularly, if all variables are independent. the Bayesian network degenerates to a Naive Bayes Model [46]

2.2.2 Markov Random Field and Ising Model

In most cases, the causality of the variables are unknown. The only structure is the correlation structure we have observed and inferred. In this case the network of variables are undirected. If the variables between the networks meets the Markov property which means there are no implied dependencies between two connected nodes, it forms a Markov Random Field (Fig. 2.6).

The coupling rules in RMF can be arbitrary. There are several coupling family in which can be solved analytically. Sparse multivariate normal distributions represented by correlation matrices are an example of the RMF.

An other example for Markov Random Field which is widely used in Physics is the Ising Model [47]. The Ising model is a discrete probability model which simulates the magnetic state of molecules in a lattice (or randomly connected spin glass). Generally, consider a node

k in a network, The state of node k is σ_k where $\sigma_k \in \{+1, 1\}$, representing the site's spin. A spin configuration, σ is an assignment of spin value to each lattice site.

For any two adjacent nodes i, j there is an interaction J_{ij}. Also the node j is driven by an external magnetic field h_j interacting with it. The energy of a configuration σ is given by the Hamiltonian function:

$$H(\sigma) = -\sum_{(i,j)} J_{ij}\sigma_i\sigma_j - \mu\sum_j h_j\sigma_j \tag{2.15}$$

where the first sum is over pairs of adjacent spins (every pair is counted once). The notation (i, j) indicates that nodes i and j are nearest neighbors. The magnetic moment is given by μ. The configuration probability is given by the Boltzmann distribution with inverse temperature $\beta \leq 0$:

$$P(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z_\beta} \tag{2.16}$$

where $\beta = (k_B T)^{-1}$ and the normalization constant

$$Z_\beta = \sum_\sigma e^{-\beta H(\sigma)} \tag{2.17}$$

is the partition function.

For the 1-D lattice (chain) or a tree of spins, Ising models are analytically solvable [47]. However, for a random network structure, as long as it maintains the properties of locally tree-like. It can be solved by belief propagation [48].

2.2.3 Boltzmann Machine and Restricted Boltzmann Machine

The Full Boltzmann Machine, or Hopfield Network with hidden nodes is a theoretical model based on fully connected networks to determine network structure and variable dependencies

at the same time. It is the one of the base of a network based machine learning model [49].

Consider a fully connected RMF with n observable variables and m hidden variables. All variables are fully connected by undirected links which refers to all possible dependencies between these variables (Fig. 2.7a). Similar to Ising model, given a configuration of variable spins $\mathbf{s} \in 0, 1^n$ the energy of the system is yield:

$$E = -\left(\sum_{i<j} w_{ij}\, s_i\, s_j + \sum_i \theta_i\, s_i\right) \tag{2.18}$$

Where w_{ij} is the connection strength between node i and node j. s_i is the state(Boolean/Bernoulli), θ_i is the bias of node i in the global energy function. Just the same as Ising model, the energy distribution of full Boltzmann Machine is:

$$P(\mathbf{s}) = \frac{e^{-E(\mathbf{s})}}{Z} \tag{2.19}$$

where the definition of Z is the same in Ising model but without parameter β

Theoretically, by training the Boltzmann machine with an adequate dataset, We are able to both infer the structure of node dependencies and making prediction on one or several of the visible nodes. However, the training algorithm for full Boltzmann machine is NP hard which makes it almost impossible in practical use [50].

Therefore a reduced model called Restricted Boltzmann Machine (RBM) is introduced for practical training [51]. Restricted Boltzmann machine assumes the independence between visible nodes and hidden nodes thus the connections between same types of nodes are removed and become a network with two layers (Fig. 2.7b).

The standard type of RBM has binary-valued (Boolean/Bernoulli) n hidden and m visible nodes. Similar to Boltzmann Machine, RBM consists of a matrix of weights $W = (w_{i,j})$ (size $m \times n$) associated with the connection between hidden nodes h_j and visible nodes v_i, as well as bias weights (offsets) a_i for the visible nodes and b_j for the hidden units. Given these, the

energy of a configuration (pair of boolean vectors) $(\mathbf{v}, \mathbf{h})$ is defined as

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{i,j} h_j \tag{2.20}$$

Similar to full Boltzmann Machine, the probability of configuration with energy $E(v, h)$ is:

$$P(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{Z} \tag{2.21}$$

where the definition of Z is the same in Full Boltzmann Machine. Because we can only observe visible nodes, the marginal probability distribution of visible nodes is given by:

$$P(v) = \frac{1}{Z} \sum_h e^{-E(v,h)} \tag{2.22}$$

To train a RBM is more practical than full Boltzmann Machine. Consider the training set of V which consist N vectors of visible nodes $\mathbf{v}$. We want to pick a set of weights and biases. Thus the average log-likelihood function of visible nodes reaches the maximum:

$$W_{i,j}, a_i, b_j = \arg\max \frac{1}{N} \sum \log P(v) \tag{2.23}$$

To optimize the parameters $W_{i,j}, a_i, b_j$, one can use the contrastive divergence (CD) algorithm. The algorithm performs Gibbs sampling and is used inside a gradient descent procedure (similar to the way backpropagation is used inside such a procedure when training feedforward neural nets) to compute weight update [51].

2.2.4 Factor Graph and Belief Propagation

More generally, if the correlation between variables are not pairwise, one can still represent the relations in a network called factor graphs. Factor Graphs are the graph consist of two

types of nodes: variable nodes and function nodes. The variable nodes are always connected to function nodes. Fig. 2.8a is an examples of factor graph. The variable nodes refers to the variables in the model while the function nodes refers to the function in coupling the variables connect to it. In factor graph, not only pairwise correlation but also the correlation of multi-body can be calculated.

Suppose in a factor graph $G(\mathbf{x}, \mathbf{f}, E)$ where the $\mathbf{x}$ denote to the variable nodes and $\mathbf{f}$ denote to function nodes. The joint probability of the variables can be expressed as:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{a \in \mathbf{f}} \psi_a(x_{\partial a}) \tag{2.24}$$

where the $\psi_a(x_{\partial a})$ is the coupling function of function node a. $x_{\partial a}$ means the function only depend on the variables attached to a. To solve the marginal probability νx_i distribution of a specific node x_i if part of the distribution in the network is known. One can use Belief Propagation (BF)[48] to solve the problem iteratively based on the concept of message passing.

Start from each edge connecting function nodes and variable nodes(i, a) (where $i \in \mathbf{x}$ and $a \in \mathbf{f}$) At the t-th iteration, we define two messages: $\nu^{(t)}_{i \to a}(x_i)$ and $\hat{\nu}^{(t)}_{a \to i}(x_i)$. The message passing from i to a is over the probability distribution of x_i thus that: $\sum \nu^{(t)}_{i \to a}(x_i) = 1$.

Messages are updated through computations at the nodes of the factor graph on the basis at the previous iterations. For a function node a with degree k, we compute the outgoing message $\nu^{(t)}_{a \to i}$ to i based on the rest $k-1$ nodes with the coupling ruled defined in the function node. The belief propagation, or sum-product update rules, are:

$$\begin{aligned} \nu^{(t+1)}_{i \to a}(x_i) &= \prod_{b \in \partial i \backslash a} \hat{\nu}^{(t)}_{b \to i}(x_i) \\ \hat{\nu}^{(t+1)}_{a \to i}(x_i) &= \sum_{x_{\partial a \backslash i}} \psi_a(x_{\partial a}) \prod_{j \in \partial a \backslash i} \nu^{(t)}_{j \to a}(x_i) \end{aligned} \tag{2.25}$$

where $\setminus$ denotes set subtraction. It is understood that, when $\partial i \setminus a$ is an empty set (leaf), $\nu_{i\to a}(x_i)$ is the uniform distribution. Similarly, if $\partial j \setminus i$ is empty, then $\hat{\nu}_{a\to i}(x_i) = \psi_a(x_a)$. A pictorial illustration of these rules is provided in Fig. 2.8b and c. A BP fixed point of these equations is a set of t-independent messages $\nu^{(t)}_{i\to a}(x_i) = \nu_{i\to a}(x_i)$ and $\hat{\nu}^{(t)}_{a\to i}(x_i) = \hat{\nu}_{a\to i}(x_i)$.

For a locally tree-like network, after t iterations, the message will eventually converge to fix point. One can estimate the marginal distribution $\nu(x_i)$ of variable i using the set of all incoming messages. The BP estimate is:

$$\nu_i(x_i) = \prod_{a\in\partial i} \hat{\nu}^{(t)}_{a\to i}(x_i) \tag{2.26}$$

In writing the update rules, we are assuming that the update is done in parallel at all the variable nodes, then in parallel at all function nodes and so on. Clearly, in this case, the iteration number must be incremented either at variable nodes or at factor nodes, but not necessarily at both.

2.3 Neural Network Based Model

Among all network based models, the family of neural networks are the most powerful tools in developing deep learning models for complicated tasks. The original idea of neural network is to simulate the neurons activities in the cortex. However, the development of neural network now is no longer limited to mimic the neuron structures and learning processes. It has developed into a set of layer-based network learning models which can handling the complicated topology structures of variable spaces.

The common training method for the neural network family models is to use gradient descend and back-propagation. The principles for back-propagation is shown in Fig. 2.9. Derivatives of Loss Function are made and propagate layer by layer according to the chain rules. When the gradient reaches the first layer, the weight will update according to the

direction of this gradient and generate a new gradient direction for next layer. Thus the update of weights will propagate back to the top layer and finishing this training cycle for the data fed.

As we discussed in the introduction, multiple techniques has been developed to make the back-propagation and gradient descend more robust and efficient. These techniques including Adams Gradient Descend [52], ReLU [53], drop-out [54] and others.

2.3.1 Universal approximation theorem

One of the base for the layer based neural network is the Universal Approximation Theorem [55, 56]. The theorem states that a feed-forward network with a single hidden layer containing a finite number of neurons, can approximate any continuous functions on compact subsets of $\mathbb{R}^n$, under mild assumptions on the activation function. The theorem thus states that simple neural networks can represent a wide variety of interesting functions when given appropriate parameters.

In mathematical terms, the theorem can be expressed as:

Let $\varphi(\cdot)$ be a non-constant, bounded, and monotonically-increasing continuous function. Let I_m denote the m-dimensional unit hypercube $[0,1]^m$. The space of continuous functions on I_m is denoted by $C(I_m)$. Then, given any $\varepsilon > 0$ and any function $f \in C(I_m)$, there exist an integer N, real constants $v_i, b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$, where $i = 1, 2, ..., N$ such that we may define:

$$F(x) = \sum_{i=1}^{N} v_i \varphi(w_i^T x_i + b_i) \tag{2.27}$$

as an approximate realization of the function f where f is independent of φ; that is,

$$|F(x) - f(x)| < \varepsilon \tag{2.28}$$

for all $x \in I_m$. In other words, functions of the form $F(x)$ are dense in $C(I_m)$.

Note that when change the hypercube I_m to any compact subset of $\mathbb{R}^m$ the theorem still holds. Moreover, in almost cases, it is not necessary to bound the $\varphi(\cdot)$ [56].

According to the universal approximation theorem, we can construct a approximation regression model called universal approximator. Fig. 2.10a shows the sketch of a universal approximator with the first n terms. The variable nodes (round) represent the approximation terms, also called hidden variables. More hidden variables will increase the accuracy of approximation but it will largely increase the number of parameters in the model and result in the problem of overfitting. The function nodes (square) is the operation in coupling different variables. The edge of network represented for the weight w_i, v_i and the bias b_i which to be updated during the training process. Fig. 2.11 shows the result of universal approximator in approximating the sine curve using $N = 10$ and $N = 50$ hidden nodes.

2.3.2 Simple static feed-forward neural network

Based on the universal approximator, if we extend the structure as shown in Fig. 2.10, a directed network classifier with single or multiple layers of neurons are constructed. In the model there are three types of layers which process the data flow sequently: Input layer, hidden layer, and output layer. For input layer, each neuron represent for the input variables correspondingly. The value will pass directly to the 1st hidden layer. Generally the components of neurons (labeled as i) in k-th hidden layer are the following:

1. The output value of a neuron: o_i^j
2. The input of a neuron $p_i^{(k)}$ is the weighted sum of output $o_j^{(k-1)}$ from all the connected neurons from last layer:

$$p_i^{(k)} = \sum w_{ij} o_j^{(k-1)} + b_i \tag{2.29}$$

The weight w_{ij} and the bias b_i can be pre-defined or trained by datasets.

3. A non-constant, bounded, and monotonically-increasing continuous function called activation function $\varphi(\cdot)$ coupling the input and the output of neurons:

$$o_i^k = \varphi(p_i^{(k)}) = \varphi(\sum w_{ij} o_j^{(k-1)} + b_i) \tag{2.30}$$

A common choice of activation function is the sigmoid function: $\varphi(z) = \frac{1}{1+e^(-z)}$. The shape of the sigmoid function is shown in Fig. 2.13a. Another common choice for activation function is an unbounded function called ReLu: $\varphi(z) = \max(0, z)$ (2.13b). Such kind of function often used in deep networks with multiple layers to avoid gradient vanishing [53].

If we use the network as a classifier, the number of the nodes in the output layer equals the of total classes. The value of output layer $z_c^{(k)} = \sum w_{cj} o_j^{(k-1)} + b_c$ represents the log-likelihood of each class. Thus the probability for each class t_c based on the input vector is:

$$t_c = \frac{e^{z_c}}{\sum_d e^{z_d}} \tag{2.31}$$

The function above also called soft-max function in which the log-likelihood is normalized by partition function $\sum_d e^{z_d}$. When inferring the class of a given sample, we choose the class with the top probability: $c = \underset{i}{\text{argmax}} y_i$

The training for a simple neural network is to back-propagate the gradient of the loss function. In a classifier, a common loss function is called a softmax cross entropy function [57]:

$$\xi(T, Y) = \sum_{i=1}^{n} \xi(\mathbf{t}_i, \mathbf{y}_i) = -\sum_{i=1}^{n} \sum_{i=c}^{C} t_{ic} \cdot \log(y_{ic}) \tag{2.32}$$

where $y_{ic} = \delta(c_i, c)$ and t_{ic} is the softmax function estimation of sample i for class c. The function is derived from the maximum entropy principles from physics. A good property of the softmax cross entropy function is that its gradient:

$$\frac{\partial \xi}{\partial z_i} = y_i - t_i \tag{2.33}$$

which makes it very simple in the back-propagating calculation.

2.3.3 Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) is a variance of the neural network used for image recognition[58]. Given an image with pixel dimension $n \times m \times$ in RGB channel, the total size of input variables is $n \times m \times 3$ which will result in a huge number of parameters for a fully connected network. However, due to the similarity and continuity of pixels, we can reduce the space of parameters by filtering the most important features inside local areas.

Like ordinary neural networks, CNN is an ensemble of layers of neurons with different types. The types of layers in the CNN are the following [58]:

1. **Input layer** INPUT layer $[n \times n \times 3]$ will hold the raw pixel values of the image, in this case an image is resized to width n, height n with three color channels R,G,B. n is a fix number for different type of models.

2. **Convolution layer** In CONV layer, we choose k different filters with the same size $l \times l$. The typical size of l ranges from 3 to 11. The filters filter the regions by direct computing a dot product between their weights and the corresponding region they are connected to in the input volume on a specific channel. By sliding the filters over

whole images, we will get a output matrix with size $[(n-l+1)\times(n-l+1)]$. Fig. 2.12a shows such process called "convolution". The weights of filters are left to be optimized during the training process. The filters will filter the particular shape of features (circles, lines, etc) according to their weights. These features are the basic components for classifications.

Because the size of output will shrink after each convolution area, sometimes we will add pads of 0 values outside the image pixels to extract the informations on edges [59]. Such process called "padding" may result in a output volume of $[n\times n\times k]$ if we decided to use k filters.

3. **ReLU layer** RELU layer will apply an elementwise activation function, such as the ReLU function $ReLU = \max(0, x)$ (Fig. 2.13) thresholding at zero. The process for ReLU layer is to filter out the meaningless area to the filter. This leaves the size of the volume unchanged while making it more sparse ($[n\times n\times k]$).

4. **Pooling layer** POOL layer will perform a downsampling operation along the spatial dimensions of (width, height), resulting in a shrink of volumes to make the variable spaces. A common approach for pooling section is to capture the max feature score by different partition regions (Fig. 2.12b). suppose we choose four partitions evenly, the resulting volume for this step is $[\frac{1}{2}n\times\frac{1}{2}n\times k]$

   The above Convolution-RELU-pooling layers are usually combined as an element agent of CNN. By apply the three steps repeatably, at the end of the network, we will get a vector indicating the only scores of each particular features. Suppose we have k' features eventually, the volume of size now become $[1\times 1\times k']$

5. **Fully Connected Layer** As the last layer of CNN, FC layer will compute the class scores, resulting in volume of size $[1\times 1\times C]$, where each of the C numbers correspond

to a class score of specific class. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous $[1 \times 1 \times k']$ volume.

Numerous variation can be applied to the network structure designs such as the output with different levels of features [1], $[1 \times 1]$ convolution on features [60], and residual networks [60]. Fig. 2.14 shows an example of inception v3 network [1] developed by Google. Many mature deep CNN architectures have been developed in general purpose of classification problems with huge amount of datasets. In many cases such pre-trained network will be used as a transfer learning [61] of topics where the number of samples are not adequate to train a network from scratch.

2.3.4 Recurrent Neural Network (RNN)

Another important type of neural network is Recurrent Neural Network (RNN) which is designed for sequence input of variables. Unlike the ordinary neural network, RNN also takes the output of same network from last time step. For the simplest RNN, it can be expressed by a feed-back loop apply to the network (Fig. 2.15a).

Recurrent Neural networks can both used as supervised learning and reinforcement learning [62, 63]. In supervised learning, sequences of input vectors arrive at the input nodes one at a time. At any given time step, each non-input unit computes its current activation output based on current input $\mathbf{x}$ and previous state output $\mathbf{s}$ just the same as ordinary neural network. Supervisor-given target activations can be supplied for some output units at certain time steps. For example, if the input sequence is a speech signal corresponding to a spoken digit, the final target output at the end of the sequence may be a label classifying the digit.

Similar to ordinary neural networks. the loss function can be chosen based on the errors produced by each step of sequence. The error for the sequence is the sum of the differences

of all target signals from the corresponding activations computed by the network. We can choose the Mean Squared Error (MSE) as the loss function to optimize.

In reinforcement learning settings, no teacher provides target signals. Instead a fitness function or reward function is occasionally used to evaluate the RNN's performance, which influences its input stream through output units connected to actuators that affect the environment. This might be used to play a game in which progress is measured with the number of points won.

A basic recurrent neural network uses the short-term memory which only considers the output of the last time step. To preserve the memories of previous time steps and avoiding gradient vanishing with the time. We change the structure of the network into Long short-term memory (LSTM) units[64] (Fig. 2.15b). A common architecture for LSTM cell is composed of a memory cell, an input gate, an output gate and a forget gate. The structure of LSTM cell including a forgot gate and coupling through the rules shown below:

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
\tag{2.34}
$$

Where

$x_t \in R^d$: input vector to the LSTM unit

$f_t \in R^h$: forget gate's activation vector

$i_t \in R^h$: input gate's activation vector

$o_t \in R^h$: output gate's activation vector

$h_t \in R^h$: output vector of the LSTM unit

$c_t \in R^h$: cell state vector

$W \in R^{h \times d}$, $U \in R^{h \times h}$ and $b \in R^h$: weight matrices and bias vector parameters which need to be learned during training.

the initial values are $c_0 = 0$ and $h_0 = 0$ and the operator $\circ$ denotes the Hadamard product(entry-wise product)[65].

Last but not the least, the direction of RNN can be changed if we are dealing with the inference require both the feed-forward and feedback information such as text understanding. In this case bi-directed RNN is introduced, Fig. 4.3 are two examples of bi-directed RNN with LSTM gate. Two sets of different weights are used in the two direction. The final output is a combination of both forward and backward output.

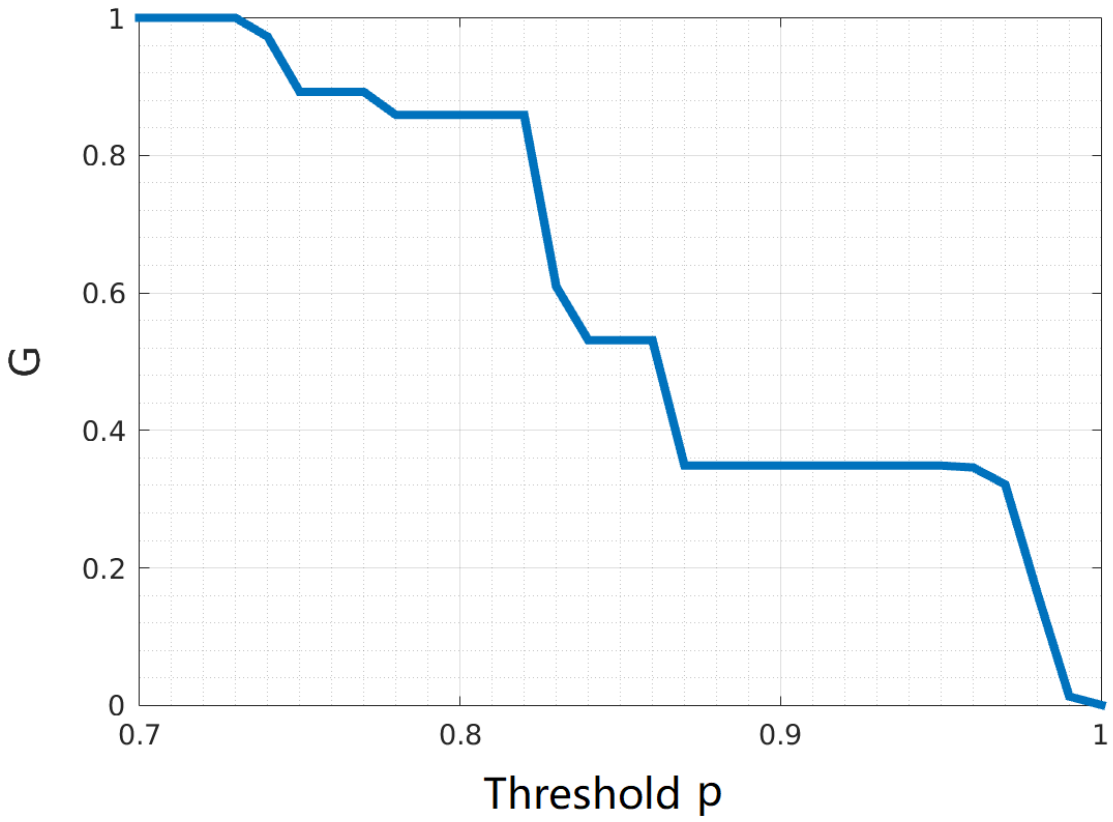

Figure 2.1: Percolation on a correlation matrix over threshold p
The example is for a vortex activity correlation matrix of a healthy subject when doing the verbal task. Courtesy from Qiongge Li.

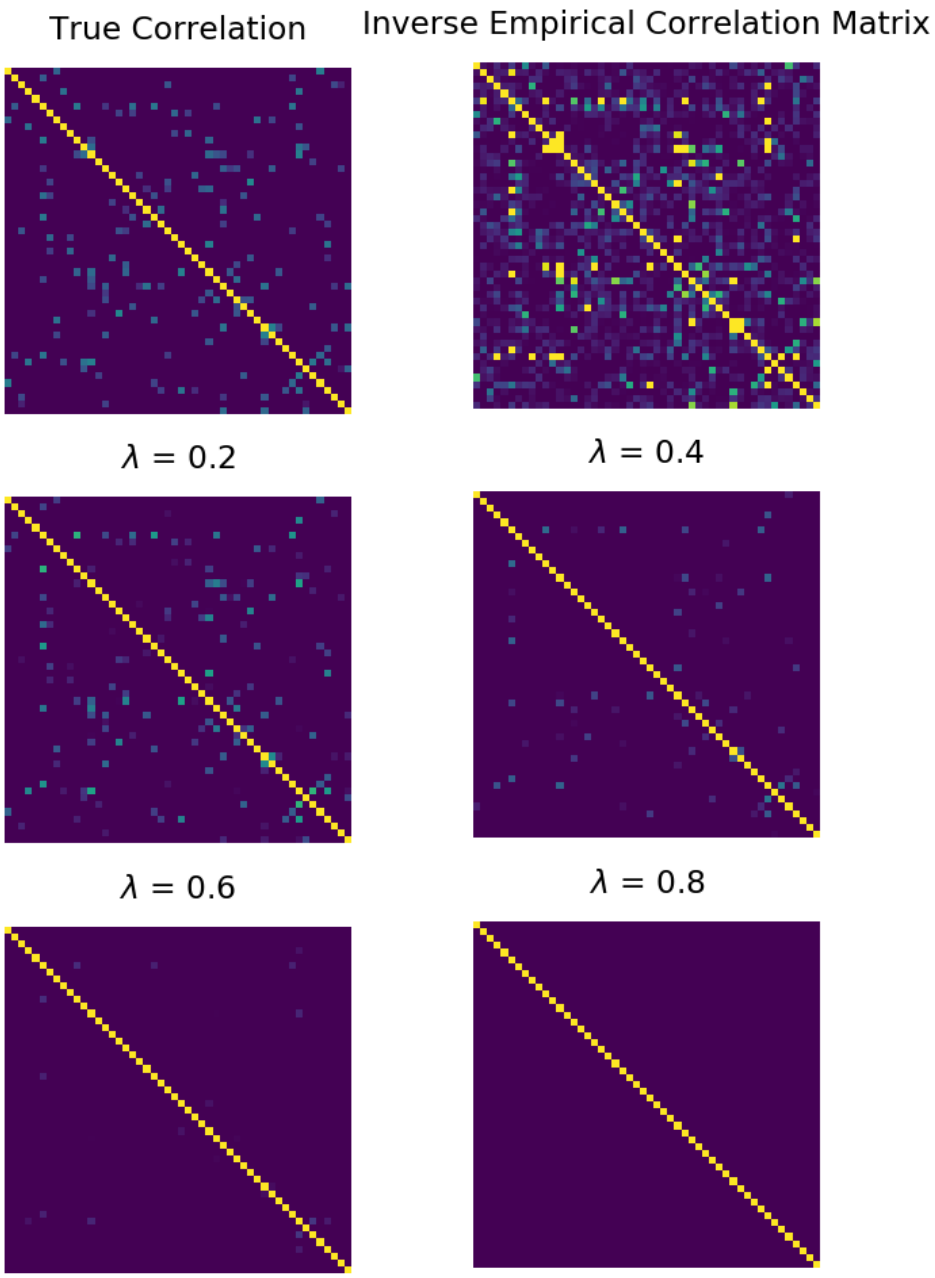

Figure 2.2: Correlation structure inference through GLASSO.
The true connection matrix $\mathbf{J}$ has 50 variables. The empirical correlation matrix is calculated from 500 samples generated from multivariate normal distribution determined by $\mathbf{J}$. The bottom 4 figures indicate the result inference $\hat{\mathbf{J}}$ under different penalty parameter λ.

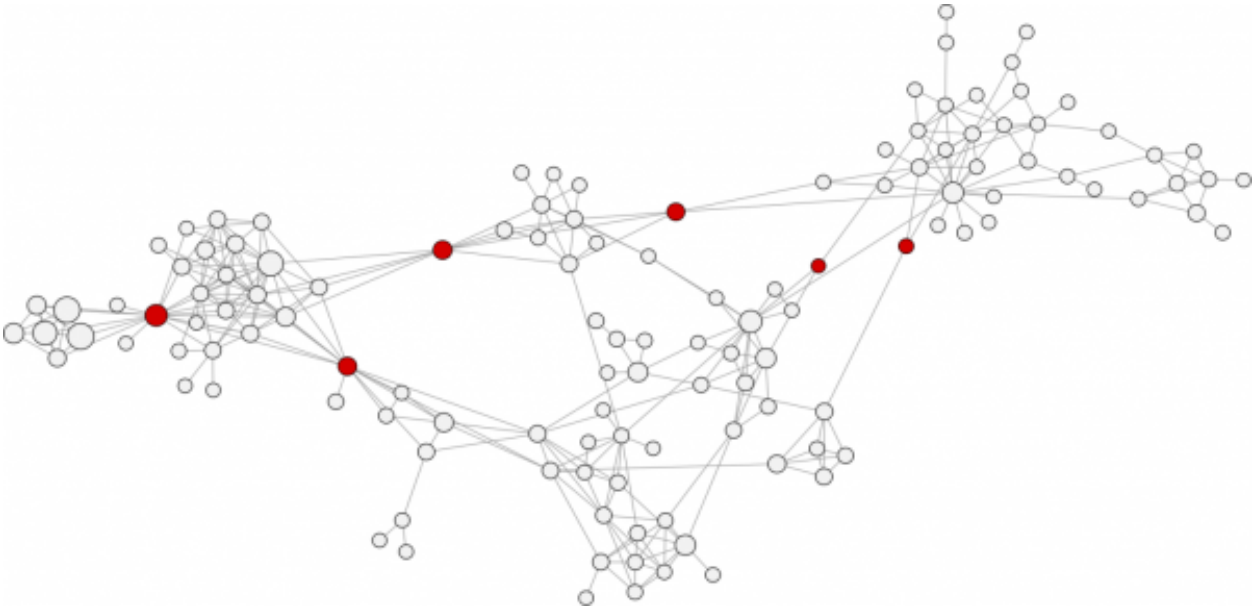

Figure 2.3: Network Betweenness Centrality
The size of the nodes are proportional to its betweenness centrality value. Particularly, the node with small degree but large betweenness centrality (cross cluster nodes) are mark in red.

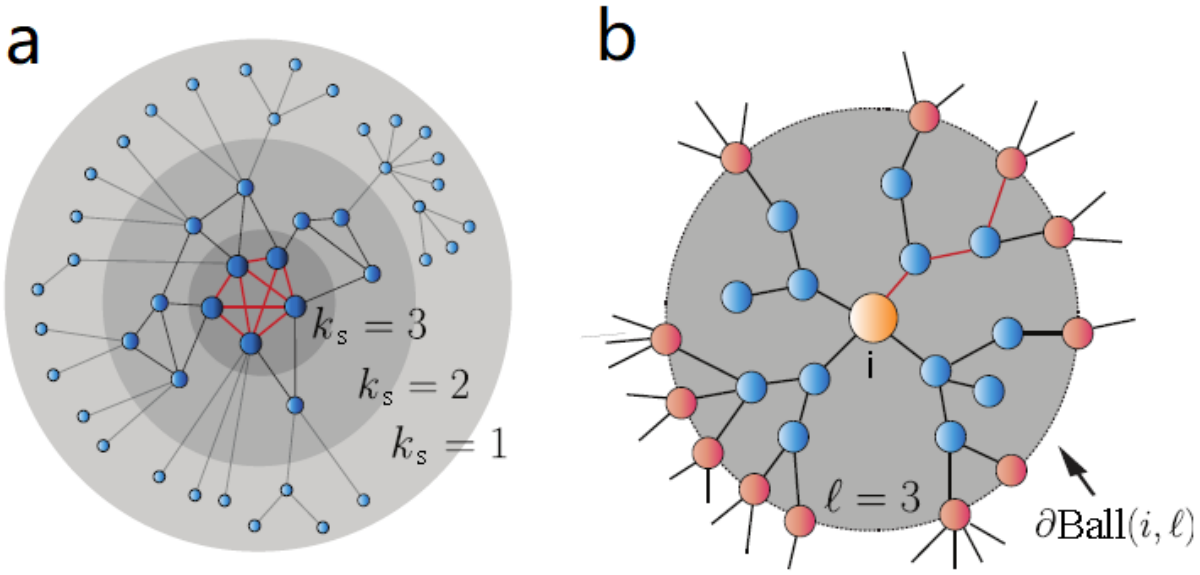

Figure 2.4: Schematic representation of k-shell and CI.
(a) Schematic representation of a network under k-shell decomposition [40]. **(b)** Example of the calculation of CI. The collective influence Ball(i, ℓ) of radius $\ell = 3$ around node i is the set of nodes contained inside the sphere and ∂Ball is the set of nodes on the boundary (brown). CI is the degree-minus-one of the central node times the sum of the degree-minus-one of the nodes at the boundary of the sphere of influence.

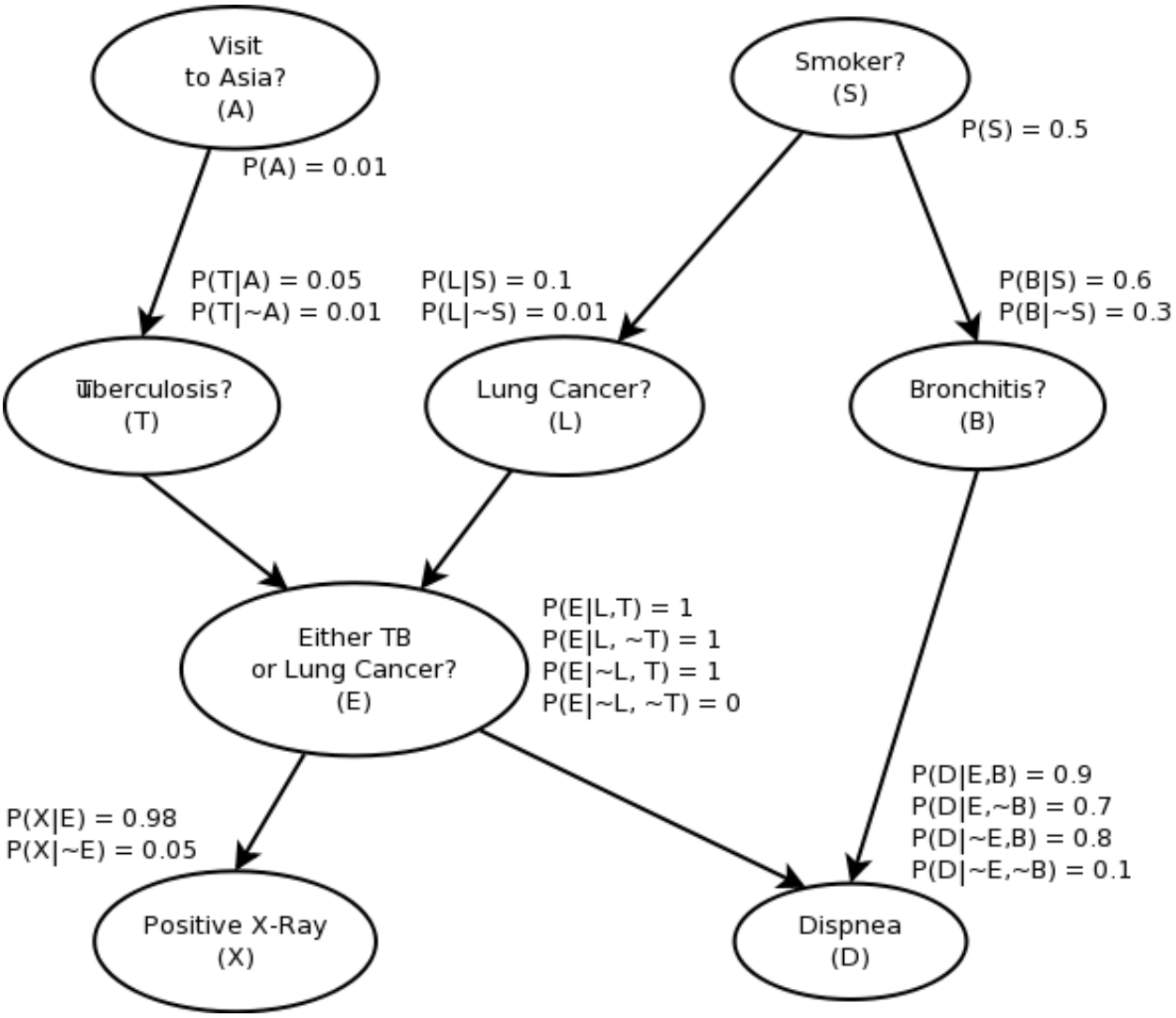

Figure 2.5: An example of Bayesian Network with probability calculation.

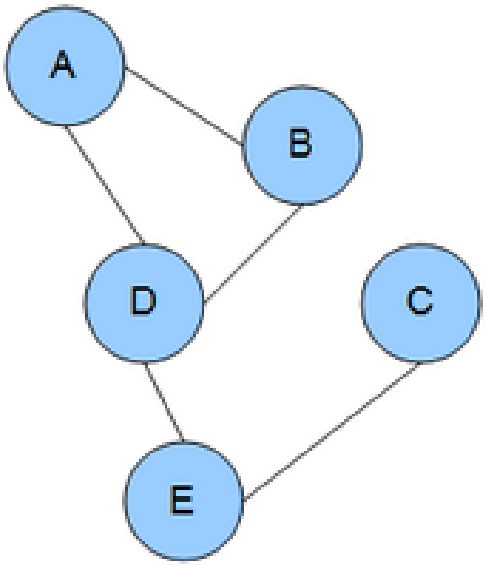

Figure 2.6: A schematic representation of Random Markov Field.

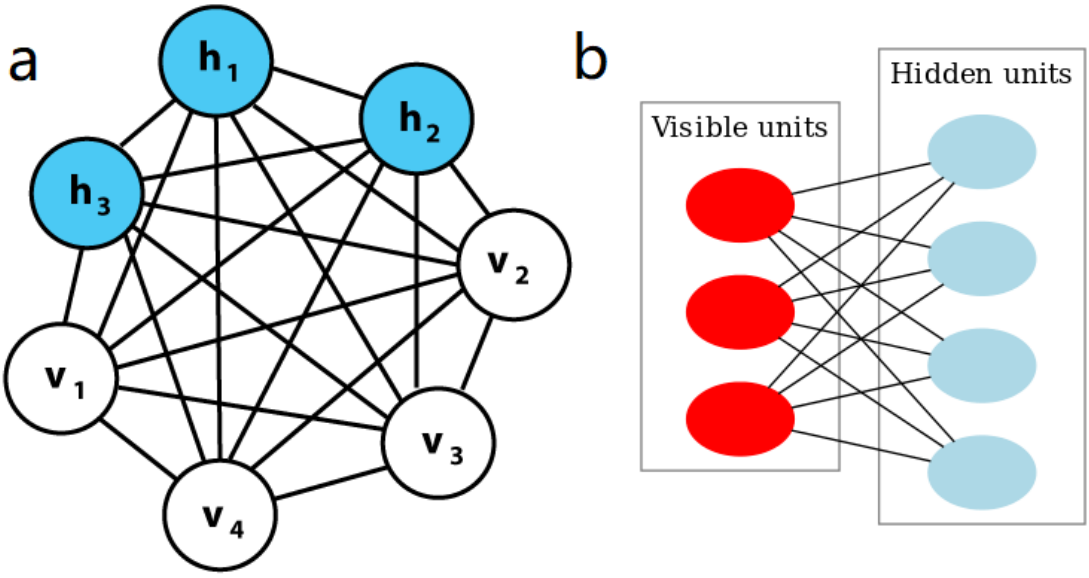

Figure 2.7: Full Boltzmann Machine and Restricted Boltzmann Machine for 4 visible nodes and 3 hidden nodes.

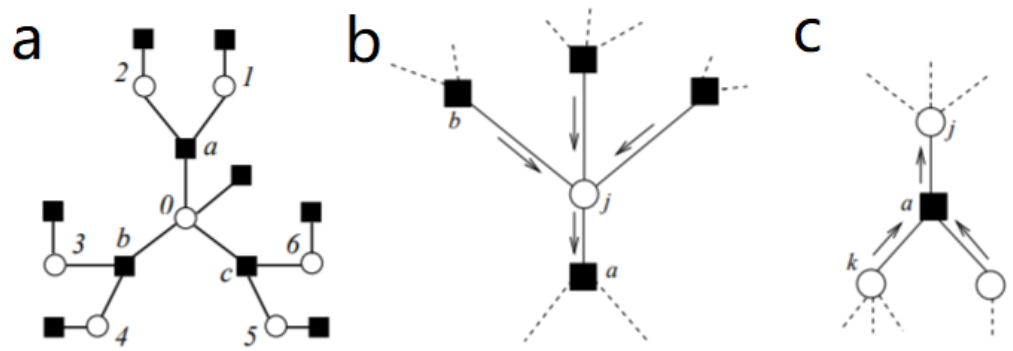

Figure 2.8: Factor graph and rules of passing message in Believe Propagation.
a) A generalized factor graph. Rules for **b)** message passing from function node to variable node. **c)** from variable node to function node.

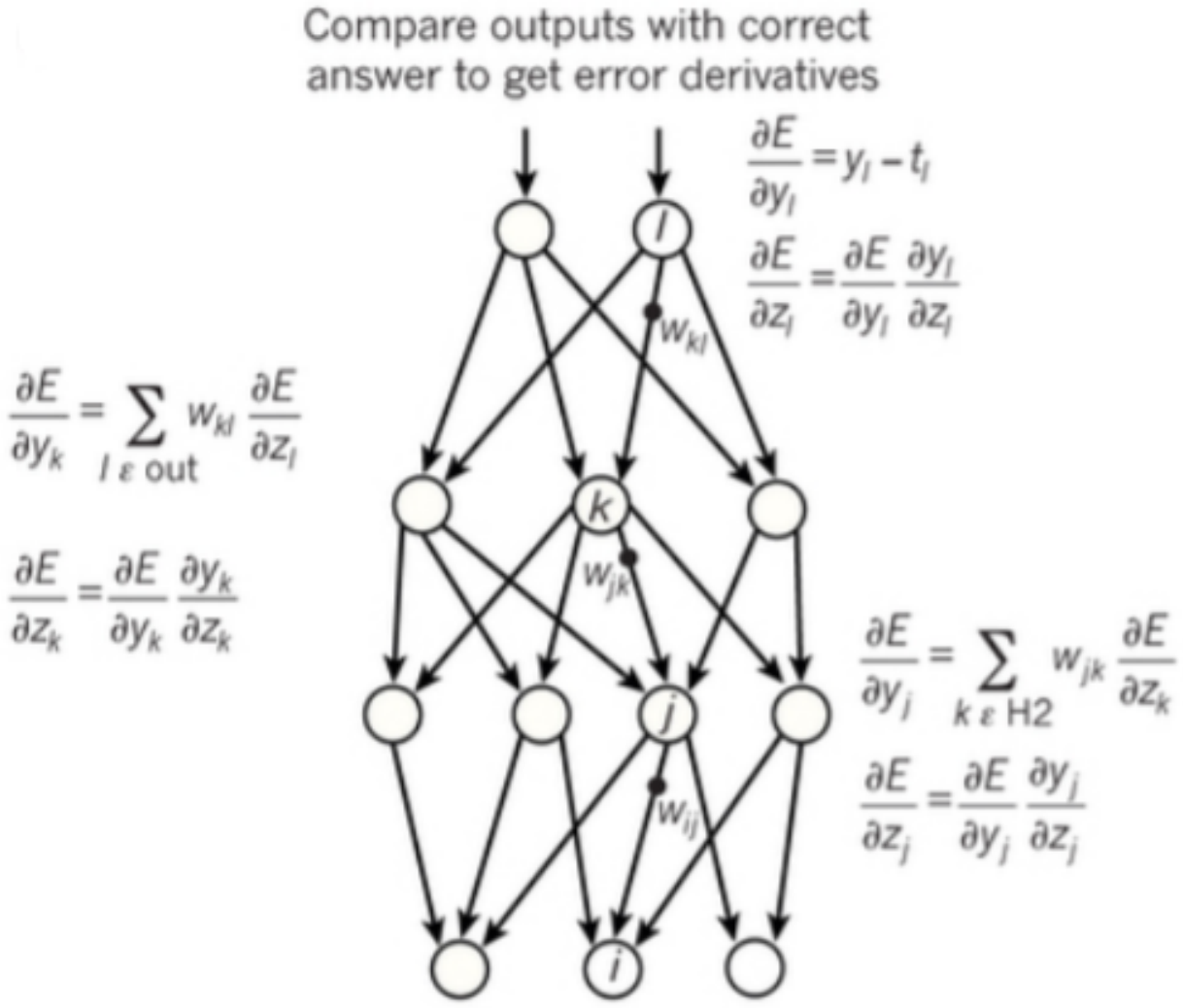

Figure 2.9: Schematic sketch of how to do back propagation using chain rule derivatives.

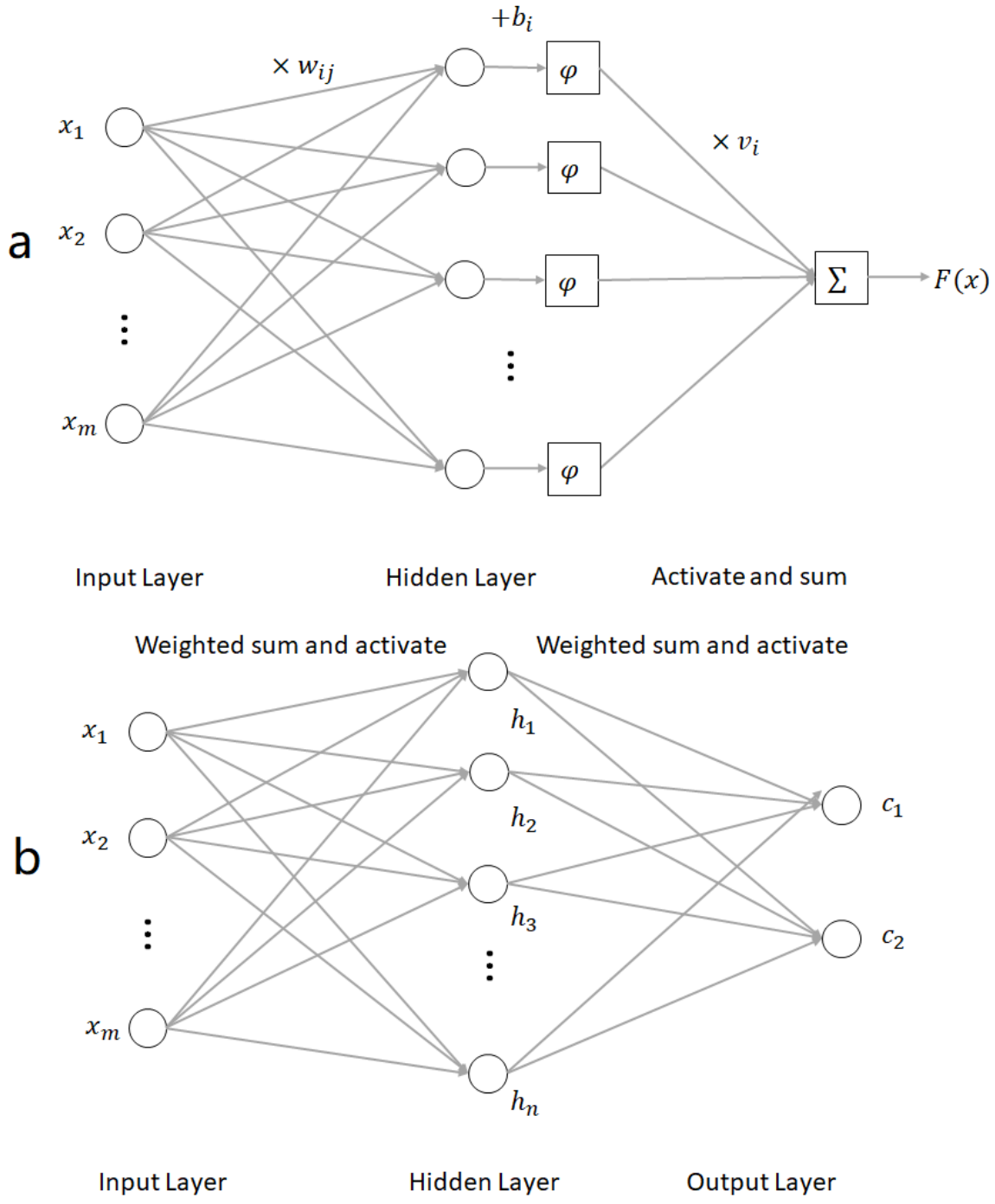

Figure 2.10: Schematic sketch of **a)** UNiversal Approximator, **a)**Basic Neural Network)

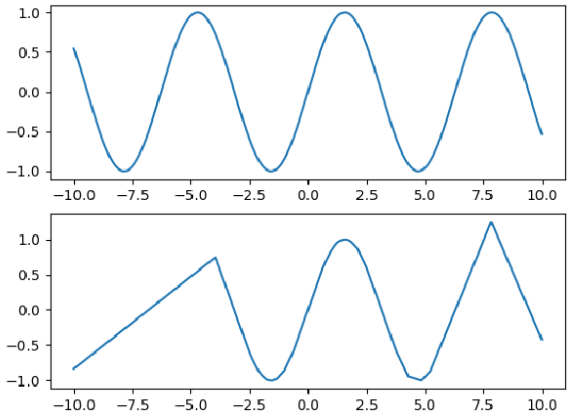

Figure 2.11: sin function approximated by Universal approximator at number of hidden nodes 50 and training step 3000

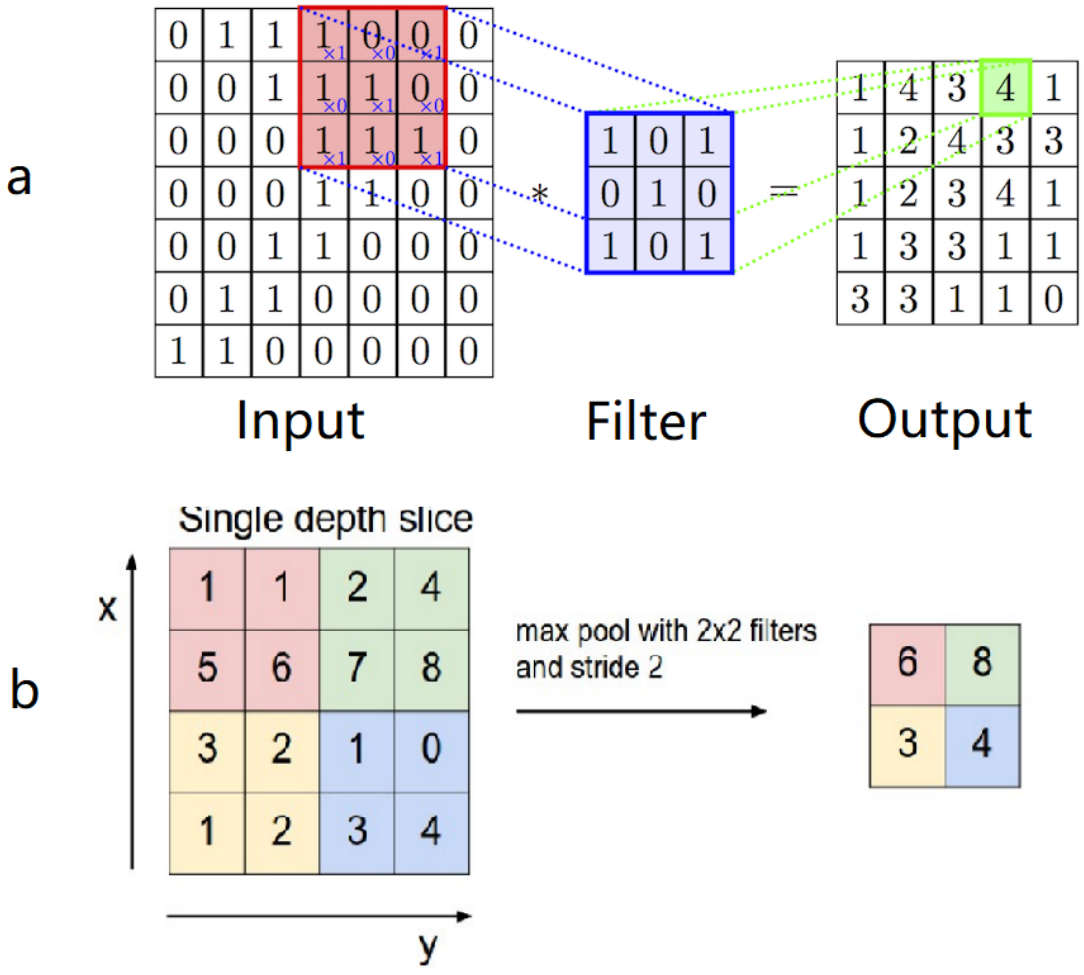

Figure 2.12: Demonstration of convolution layer and max-pooling layer in CNN

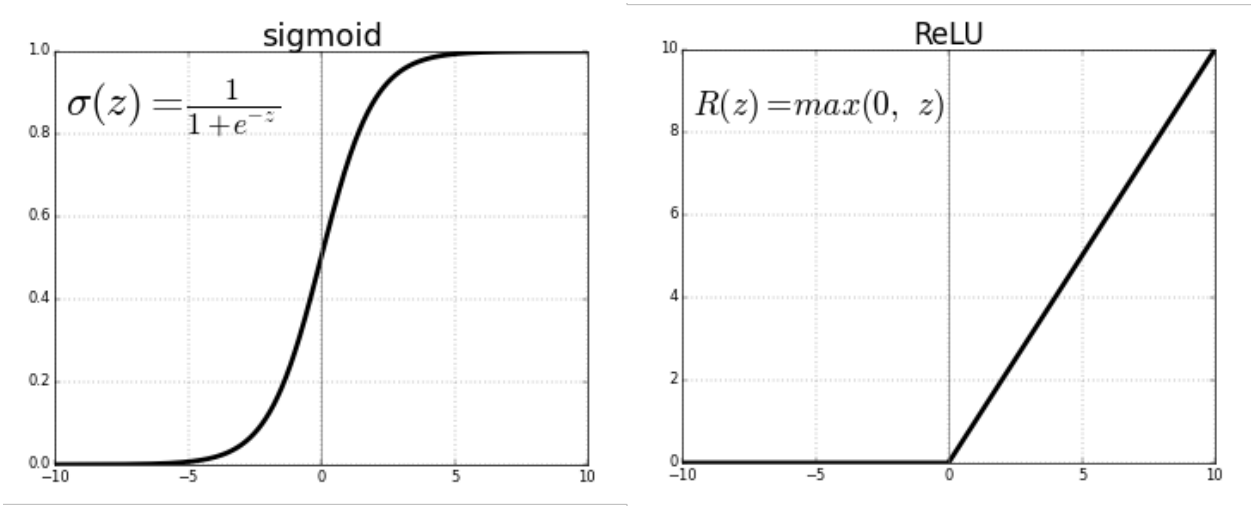

Figure 2.13: Shape of sigmoid and ReLU activation function

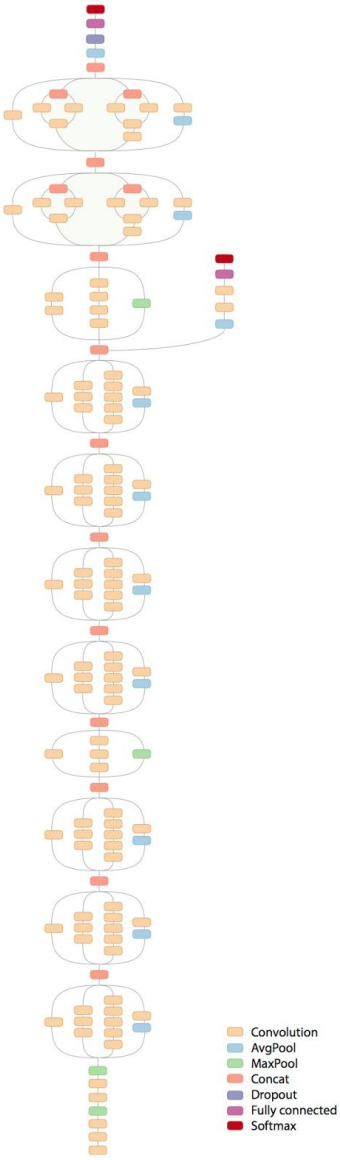

Figure 2.14: Architecture of Deep Convolutional Neural Network: Inception v3 [1]

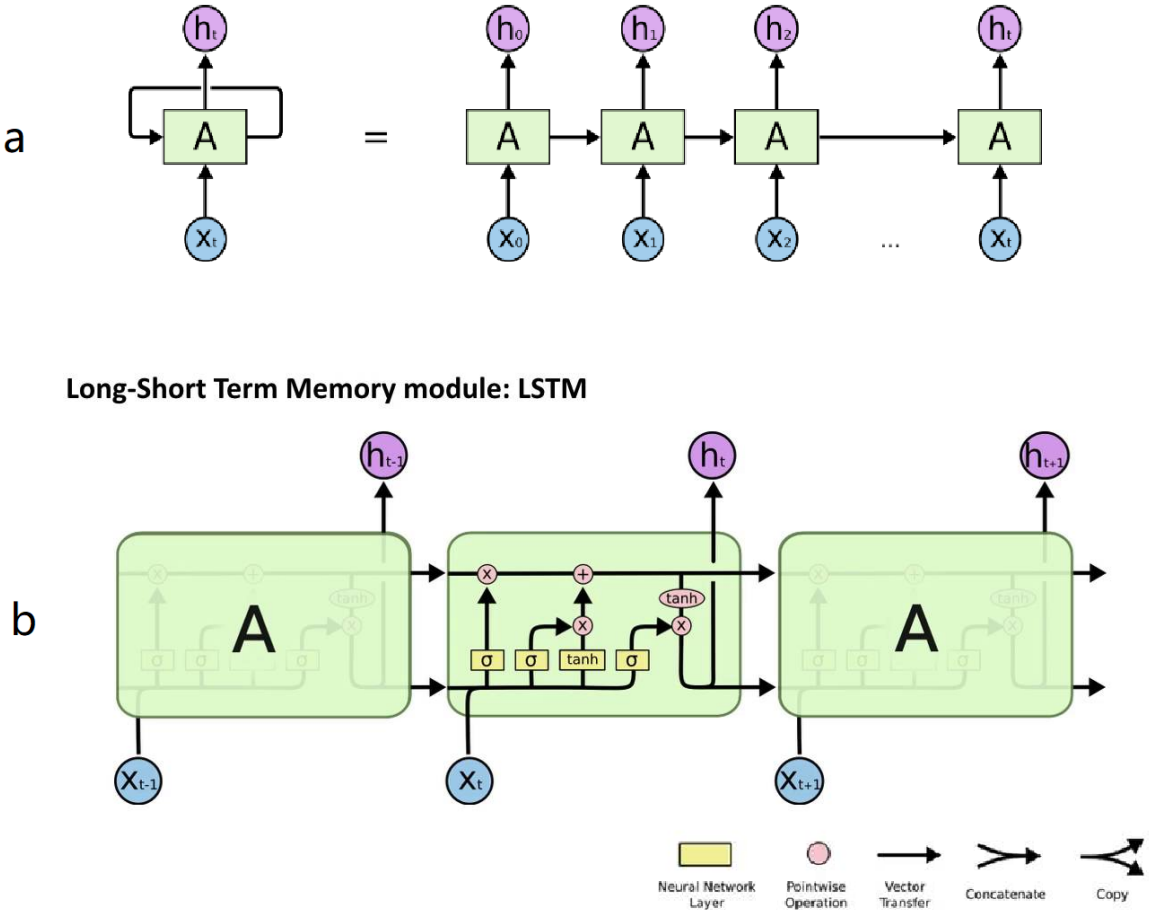

Figure 2.15: Recurrent Neural Network and LSTM cell

Chapter 3

Application 1: Inferring socioeconomic status via network location

In this chapter, we introduce the application of machine learning techniques used on a network models. We demonstrate a combination of network algorithms in evaluating the locations of individuals in social network and quantitatively verify the positive relation between network influence and personal socio-economic status. This work has been published as "Inferring personal economic status from social network location" at Nature Communications in 2017 []

3.1 Problem Description

The long-standing problem of how the network of social contacts [66, 67, 68] influences the economic status of individuals has drawn large attention due to its importance in a diversity of socio-economic issues ranging from policy to marketing [69, 70, 45, 71]. Theoretical

analyses have pointed to the importance of the social network in economic life [70] as a medium to diffuse ideas [72, 73] through the effects of "structural holes" [74] and "weak ties" in the network [69]. Likewise, research has recognized the positive economic effect of expanding an individual's contacts outside their own tightly connected social group [66, 75, 76, 77]. While previous work has established the importance of social network influence to economic status, the problem of how to quantify such correspondences via social network centralities or metrics [68, 78] remains open.

Studies employing mobile phone communication data and other social indicators have found a variety of network effects on socio-economic indicators such as job opportunities [79, 80], social mobility [81, 82, 83], economic development [45, 84, 85, 86] and consumer behavior [87, 88]. Recent work also provides evidence of such effects on an individual's wealth, and highlights the need for better indicators [89]. Recently, a numerical study has tested the effect of network diversity on economic development [45]. This study analyzed economic development defined at the community level. However, the question of how social network metrics may be used to infer financial status at the individual level—necessary, for instance, for micro-target marketing or social intervention campaigns—still remains unanswered. The difficulty arises, in part, due to the lack of empirical data combining an individual's financial information with the pattern of their social ties at the large-scale network level of the whole society.

3.2 Data Description

In this work, we address the above problem directly by combining two massively-large datasets: a social network of the whole population of a Latin American country and financial banking data at the individual level. The present framework and data acquisition have gone through an extensive process of revision and approval that took more than one

year and have IRB approval Protocol No. 2016-1418 at City University of New York. In the framework of the study, the private and/or sensitive information of the telecommunications company clients was protected. In particular, the Bank didn't gain access to any individual information about the telecommunications company users. Similarly, the private information of the Bank's clients was protected in the framework of the study. In particular, the telecommunications company didn't have access to the individual information of the Bank's clients. The variables shared were revised to guarantee that the privacy of clients was protected.

All of our datasets are encrypted and securely stored. The mobile dataset consists of records of phone calls and SMS (short message service) metadata which was collected from clients of a major operator of a Latin American country. The dataset is anonymized. All the data are encrypted and stored in a server secured by enterprise-grade firewall. The records cover a period of 122 consecutive days. Each phone number was encrypted by a high level of hashing in order to eliminate all possible access to personal information. For our purposes, each CDR (Call Detail Record) is represented as a tuple $\langle$x, y, t, dur, d, l$\rangle$, where x and y are the encrypted phone numbers of the caller and the callee, t is the date and time of the call, dur is the duration of the call, d is the direction of the call (incoming or outgoing, with respect to the mobile operator client), and l is the location of the tower that routed the communication. Similarly, each SMS metadata record is represented as a tuple $\langle$x, y, t, d, l$\rangle$. We constructed a social network $G = (N, E)$ based on the phone call and SMS traffic. Both reciprocal and non-reciprocal links are preserved for further processing.

In inferring the real social network from the mobile network, we take the assumption that the communication demands are rigid against the cost, which is usually affordable to most families ($\sim$USD \$17 monthly cell phone service fee vs. $\sim$USD \$600 monthly income in the year data was collected, respectively). Thus, the direct impact of an individual's financial status on the communication structure evidenced in the mobile phone network might be limited. However, the financial cost of using phone services makes it possible that there is

a systematic bias in how much wealthy individuals use the phone services relative to people that have less money to spend on phone calls. At this point, with the present data, we cannot rule out this possibility.

The financial dataset from a major bank in the same country was collected during the same time period as the mobile dataset. These data record financial details of 1.23×10^6 clients assigned unique anonymized identifiers over the same three-month period as the mobile network. The dataset consists of records of the bank clients' age, gender, credit score, total transaction amount during each billing period, credit limit of each credit card, balance of cards (including debit and credit), zip code of billing address, and encrypted registered phone number. A subset of 5.02×10^5 clients have an encrypted mobile phone number, thus enabling them to be matched with the mobile communication dataset. The phone numbers are encrypted in the same way as in the mobile dataset, which guarantees that the two datasets are matched. Excluding the information on credit lines, all other personal information is erased. We sum up the credit limits of all the credit cards of each account owner to represent the total credit limit of each individual.

In the absence of direct access to an individual's income and total assets, evaluating an individual's financial status remains an open question. In this dataset, we can access the following factors:

Transaction amount, which also directly reflects the individuals' consumption patterns. However, since it is common that one holds multiple accounts in different banks, and some of these may not be used at all, records in only one bank might not correctly reflect the real spending ability of an individual. Similar reasoning can be applied to total credit card balance per month, which could also lose its ability to measure one's financial status.

Credit scores assigned to individuals by credit scoring agencies are also good indicators of financial status. However, the values of credit scores are quite limited, ranging from 300 to 850. This limited range makes the credit score a low-resolution indicator of wealth that does

not allow us to correctly classify a large number of people into well-defined financial classes. On the other hand, the credit limit ranges over three orders of magnitude, allowing us to correctly classify the entire population. Considering the weaknesses of the other features, total credit limit is the most convenient measure of personal financial status in the present dataset.

Instead of transaction amounts and credit scores, we choose the total credit limit which is assigned by the bank after comprehensive evaluation of an individual's financial status, as a proxy for financial status. Since detailed information on how the credit limit is assigned is not provided, there are several possible factors that could cause bias in inferring an individual's real economic status. These include the delay of credit limit in reflecting a change in an individual's financial status, possible correlation with the age of the account, and so on. In fact, the credit limit might be capturing the amount of information the bank has about the customer, instead of his/her actual income.

3.3 Construction of the network

The social network was constructed based on the phone call and SMS traffic. Both reciprocal and non-reciprocal links are preserved. However, as we known, individuals daily contacts included a varies types of phone lines including business lines and spam spreaders. Before we performed any further analysis on the network, we need to remove such lines because they may largely affect the structure of network.

3.3.1 Anomaly removal based on mobile network behavior

Inferring social network structure through mobile phone data requires the removal of lines operated by non-humans. Due to privacy restrictions, we could not filter business landlines and spawn spreaders at the outset. Several ways of filtering the landlines were applied in

previous works, including setting a cut-off threshold degree [45] or only considering reciprocal phone calls [90]. However, these methods usually also cut off some important human communication behavior in that particular window of observation. All communication events should be considered in evaluating the social network. Therefore, the key problem is to find a method to distinguish human- and non-human-operated lines while retaining maximal information about individuals' communication patterns.

Although we do not have the human/non-human label for the totality of the phone lines, which could separate at the outset the non-human-operated lines, we are in possession of the set of phone numbers registered with the bank dataset. These human-operated lines provide the possibility of supervising a machine learning process to learn the human behavior that separates them from robots and non-human-operated lines. We set up a hypothesis test by modeling the human-operated lines based on several variables. We first cluster the human-operated lines in a hyperspace. A new unlabeled node will be assigned a p-value according to its distance to the cluster. By carefully choosing a threshold of the p-values, we can label the node according to whether we accept or reject the hypothesis that the line is operated for personal use.

A training set consisting of the phone lines in the bank database (1.23×10^6 nodes), which is around 1% of all of the data in the entire network (1.10×10^8 nodes), was set up. We define a call or message from phone number i to j as a 'communication event,' and denote the total number of communication events on the link as $W_{i \to j}$. The key assumptions of the model are the following:

1. Communication between lines of personal use is usually (but not always) reciprocal. This means that the fraction of paired communication events on human-operated lines is generally higher than that of unpaired ones. Namely, it suggests that although communication

load difference D_i on every line:

$$D_i = \left| \sum_{j \in \partial i} W_{i \to j} - \sum_{j \in \partial i} W_{j \to i} \right| \tag{3.1}$$

should increase with degree k, it should be bound by an upper limit in the case of human-operated lines. Numbers operated for non-personal use like business hubs and spawn spreaders may have very large D_i because they are usually operated only for sending or receiving phone calls independently, but not for both at the same time.

2. Other types of business hubs may have large numbers of paired communications despite their limited D_i. These business hubs include the phone numbers for company landlines, roadside assistance, or other services requiring instant follow-up by the recipient of the phone call. To filter out these hubs we assume that the paired communication:

$$R_i = \sum_{j \in \partial i} \min(W_{i \to j}, W_{j \to i}) \tag{3.2}$$

also increases with k, but is limited for lines for personal use. The decay of the tail is supposed to follow a power-law due to the preferential attachment rule [90].

The last assumption is: 3. Most phone numbers in the network are for personal use, which results in the number of non-human-operated lines being small.

After we introduce these basic assumptions, empirical analysis can be applied to build a model describing human-operated line behavior. The model simplifies to a parametric probability distribution depending on two random variables D_i and R_i, and a variable maximum degree k which controls the parameters. Under the preferential attachment rule of assumption 2, it is reasonable to assume the distributions of both D_i and R_i for a given k deviate from a maximum entropy distribution and show a power-law tail. A good approximation is

the log-logistic distribution:

$$P(D_i|k) \sim LL(d_i, \alpha_{\mathrm{D}}(k), \beta_{\mathrm{D}}(k)), \tag{3.3}$$

and

$$P(R_i|k) \sim LL(r_i, \alpha_{\mathrm{R}}(k), \beta_{\mathrm{R}}(k)), \tag{3.4}$$

where

$$LL(x, \alpha(k), \beta(k)) = \frac{(\beta/\alpha)(x/\alpha)^{\beta-1}}{[1+(x/\alpha)^{\beta}]^2}. \tag{3.5}$$

This also suggests the logarithm of both metrics follows a normal-like but exponential tailed logistic distribution:

$$P(\log D_i|k) \sim L(d_i, \mu_{\mathrm{D}}(k), s_{\mathrm{D}}(k)), \tag{3.6}$$

and

$$P(\log R_i|k) \sim L(r_i, \mu_{\mathrm{R}}(k), s_{\mathrm{R}}(k)), \tag{3.7}$$

where

$$L(x, \mu(k), s(k)) = \frac{1}{4s(k)}\mathrm{sech}^2\left(\frac{x-\mu(k)}{2s(k)}\right), \tag{3.8}$$

with $\mu(k) = \log(\alpha(k))$, and $s(k) = \frac{1}{\beta(k)}$. Based on the knowledge we have, this distribution is the best choice even though we cannot precisely provide an exact fitting. However, the fitting results strongly support the approximation geometrically (Figure 3.1). The model involves four parameter sequences: $\hat{\mu}_{\mathrm{D}}(k)$, $\hat{s}_{\mathrm{D}}(k)$ and $\hat{\mu}_{\mathrm{R}}(k)$, $\hat{s}_{\mathrm{R}}(k)$. To determine the function of dependency, we pick the interval $k = 40$ to 160. We consider this a normal range of degrees wherein the nodes are almost all human-operated to fit the trend of μ and s. Adequate numbers of observers in each degree division guarantee the reliability of the results. The estimated $\hat{\mu}_{\mathrm{D}}(k)$, $\hat{s}_{\mathrm{D}}(k)$ and $\hat{\mu}_{\mathrm{R}}(k)$, $\hat{s}_{\mathrm{R}}(k)$ can be simply described by linear models within

this range (Figure 3.2, $R^2 > 0.98$). The relations are then used to predict parameters under other degree ranges.

After validating the assumptions, we are able to implement the learning process by performing a hypothesis test:

1. Fit the model of training data and get the sequence of estimated $\hat{\mu}_{\mathrm{D}}(k)$, $\hat{s}_{\mathrm{D}}(k)$, $\hat{\mu}_{\mathrm{R}}(k)$, and $\hat{s}_{\mathrm{R}}(k)$.

2. For each node i with given difference d_i, number of communication pairs r_i and degree k_i, calculate the p-value of $p_{\mathrm{D}}(i) = P(D < d_i|k_i)$, and $p_{\mathrm{R}}(i) = P(R < r_i|k_i)$.

3. Set a threshold p using the following test to classify the nodes:

If:

$$p < p_{\mathrm{D}}(i) < 1 - p \quad \wedge \quad p < p_{\mathrm{R}}(i) < 1 - p \tag{3.9}$$

then i is a human-operated line. Otherwise a p-value outside the range defined above will be rejected by the null hypothesis: $H_0 \rightarrow i$ is a human-operated line. It will be labeled as a non-human-operated business hub due to its extraordinarily unbalanced communication pattern or large volume of communication events.

Last but not least, the threshold p should be optimized. Suppose the network follows the exact distribution given by the model above. The fraction of outliers (non-human-operated lines) ϵ is exactly $2p$. The difference $\epsilon - 2p$ can be approximately regarded as the number of non-human-operated lines or 'outliers'. Figure 3.3 is the plot of p over $\epsilon - 2p$. A maximum is reached when $p \sim 1.6 \times 10^{-5}$. At that point, the filter is the most sensitive to detecting outliers since it covers the boundary of human- and non-human-operated nodes.

The result of data filtering is shown in Figure 3.4. The final network has 1.07×10^8 nodes (97.27% of the total data) and 2.46×10^8 links. There are 4.51×10^7 reciprocal social ties. The size of the giant connected component is 99.2% and the average degree is 4.7. The maximum degree k is 1056 and the maximum total communication load of a single node

is $\sim 10K$ including messages and calls, which is reasonable for a person who is active in business contacts during a three-month period.

3.3.2 Network overlook

After filtering the non-human active nodes by a machine-learned model trained on human natural communication behavior, we construct a final network of 1.07×10^8 nodes in a giant connected component made of 2.46×10^8 links. The ties, or links, in the network correspond to phone call communications, since we expect that communication patterns are indicative of an individual's location in the social network [90, 91, 92]. The financial cost of using phone services makes it possible that there is a systematic bias in how much wealthy individuals use the phone services relative to people that have less money to spend on phone calls. Although the effect might be limited, we cannot rule out this possibility with the present data.

Despite the large scale of our data source, we note that working on a single specific country as in the present study is not enough to grant generality to our results. In order to test the general validity of the present results, access to other countries' whole-population-level communication and banking datasets would be needed. As more datasets become available, the generality of our results can be tested across different economic and social systems.

3.4 Network Influence and Financial Status

To address the problem in describing the relations between network influences and financial status, we had to carefully choose the proxies in identifying both network influences and financial status. In this section, we introduce how we obtained and defined these two concepts by regularize both financial and network metrics. Linear regression models were also performed in identifying the significances of the variables.

3.4.1 Visualization Studies

Figures 3.5a and 3.5b show the communication patterns geolocalized across the country of individuals in the top 1% and bottom 10% of credit limits, respectively. The inequality in the patterns of communication between the top economic class and the lowest is striking and mimics the economic inequality at the country level [93]. It is visually apparent that the top 1% (accounting for 45.2% of the total credit in the country) displays a completely different pattern of communication than the bottom 10%; the former is characterized by more active and diverse links, especially connecting remote locations and communicating with other equally affluent people. Particular examples of the extended ego-networks for two individuals (with same number of ties) ranking in the top 1% and bottom 10% provide a zoomed in picture of such differences (Figs. 3.5c and 3.5d, respectively). The wealthiest 1-percenters have higher diversity in mobile contacts and are centrally located, surrounded by other highly connected people (network hubs). On the other hand, the poorest individuals have low contact diversity and are weakly connected to fewer hubs.

In order to quantitatively describe the structural differences between people with different levels of credit limits, we performed an entropy analysis. First, we choose people within the top 5% and bottom 5 to 10% credit limit percentiles, representative of the wealthy and poor populations respectively. Then, we randomly divided both groups into 20 small subgroups where each subgroup contained $N(0) \sim 2700$ bank clients. Next, we expanded each subgroup's contacts by a distance ℓ to get a subnetwork and clustered the nodes in the subnetwork through modularity analysis (see chapter 2) into different communities, finally counting the number of nodes inside each community (n_i). The entropy of this subnetwork is defined as:

$$S = -\sum_i p_i \log p_i, \tag{3.10}$$

where $p_i = \frac{n_i}{\sum_i n_i}$ is the fractional size of community i. Also, we introduced two indicators: (1) $R_{\mathrm{n}}(\ell) = N(\ell)/N(0)$, which is the ratio between the size of the augmented network $N(\ell)$ and the size of the initial subgroup $N(0)$, and (2) $R_{\mathrm{c}}(\ell) = C(\ell)/C(0)$, where $C(\ell)$ is the number of communities in the augmented network and $C(0)$ is the number of communities in the initial subgroup. Supplementary Table 3.1 shows the results of entropy S, $R_{\mathrm{n}}(\ell)$ and $R_{\mathrm{c}}(\ell)$ across an average of 20 subgroups, with uncertainties.

The entropy in subnetworks generated from the poor population is higher than in subnetworks generated from the wealthy population, while the numbers of both the total communities and nodes are smaller. This suggests that the sizes of the communities in the subnetwork of poor people are relatively more balanced than in the wealthy population. Namely, wealthy people are more likely to form larger and more closely-connected communities which result in relatively low entropy. The result of R_{n} and R_{c} shows the significant difference between the size and diversity of the subnetworks of the wealthy and poor populations. By expanding their contacts, people with higher credit limits 'collect' more people and more communities. Such differences exist even when we increase the value of ℓ to 4. The result of the entropy analysis implies that the network structure of these two groups may be significantly different. Wealthy people have higher diversity in mobile contacts and are centrally located, surrounded by other highly-connected people (network hubs).

Entropy analysis results also provide evidence of homophily, which implies that there exists a higher probability that two wealthy individuals are connected than that a wealthy individual and an extremely poor individual are connected. Since society is known to have this strong stratification property embedded in social networks, we would expect that this feature is expressed in our network. For example, if wealth implies higher degree, then homophily will lead to angle correlations, higher k-shell scores for wealthy individuals, and higher CI. Thus, part of the effect we observe in the present study might be due to the effects of homophily. However, the exact picture of how homophily affects the wealthy population

is still to be discovered. The crux of the matter is to find a reliable social network metric to quantify this visual difference in the patterns of network structure between the rich and the poor, as we show next.

3.4.2 Identifying personal financial status

As discuss in the first section, financial status is obtained from the combined credit limit on credit cards assigned by banking institutions to each client. The credit limit is based on composite factors of income and credit history and therefore reflects the financial status of the individual. The credit limit is pulled from an encrypted bank database and identified by the encrypted clients' phone numbers registered in the bank. Thus, we are able to precisely cross-correlate the financial information of an individual with their social location in the phone call network at the country level. There are 5.02×10^5 bank clients who have been identified in the mobile network whose credit limit ranges from USD \$50 to $\$3.5\times10^5$ (converted from the country of study). Thus, the datasets are precisely connected providing an unprecedented opportunity to test the correlation between network location and financial status.

We use the following statistics to identify economic effects: First, we separate the individuals into groups on sampling grids in variable space (1D as segment bins and 2D as grids). In each group (with more than 10 people for statistical significance), we count the fraction of wealthy individuals, defined as those individuals in the top 4-quantile $Q > 0.75$ or who have a total credit limit greater than USD \$4,000 (converted).

Besides the credit limit, transaction amount and credit score the bank data also provides the information of the clients' birth years. Age as a variable is independent from the network metrics (Supplementary Table 3.2) and correlates with the percentile-ranking credit limit ($r = 0.42$). However, we do not know the model used by the bank to assign the credit limit, so the age may be a complex reflection of the mixed effects of both increased income and

increased account history. Thus, the correlation between age and credit limit might not be capturing only variation in actual wealth but also the amount of information the bank has about the customer.

To quantitatively evaluate the variance caused by network metrics when combined with other factors, we employed Analysis of Covariance (ANCOVA) [94]. ANCOVA is an analysis method which conducts regressions between covariate (CV) and dependent variables (DV) under different groups of categorical independent variables (IV). In this case, regression was made between covariate CI and the dependent variable, the fraction of wealth. As in Fig. 3.11d, CI is divided into 100 partitions. Based on the information to which we have access, ANCOVA was applied separately among the following independent variables: gender, age, and residential communities. Gender was naturally divided into two groups. Age was grouped year by year from 18 to 65 in a total of 48 groups. The communities were identified by their registered zip code. To reduce the dimensionality of the problem and directly quantify the effect of geographical location, we first sorted the communities by the fraction of wealthy people inside and divided them into 50 balanced groups. We assigned to every community an 'Index of Community Wealth' (ICW), which is the quantile ranking of each group that the community belongs to.

The correlation between IVs and CV are shown in Supplementary Table 3.3. The negligible correlation between these variables ensures the basic assumption of independence in ANCOVA. Also, in order to test the robustness of our results, the same method was applied under different thresholds of credit limits to define the wealthy population: Q = 0.75 (the threshold we used), 0.85 and 0.95.

The basic output of ANCOVA is a series of p-values showing the significance level of the regression model between CV and DV in different IV groups, and the analysis of variance (ANOVA) [94] evaluating the significance of the IVs' effects. The estimated slopes with 95% confidence intervals are shown in Figure 3.7. Our results show the following:

1. All IVs' effects are significant ($p < 0.001$); namely, the fraction of wealthy people is different among different groups of gender, age or communities.

2. Inside most groups of each IV, the variation caused by CI is also significant ($p < 0.001$). The only exception is that CI's effect is only significant when the clients are older than 24 years (Supplementary Figure 3.7b). This result indicates that the effect of network metrics, in most cases, is independent from the other known factors.

3. The slope of regression varies in different groups. However, all the slopes with significant values are positive.

4. The results of 1 to 3 above are robust under different thresholds of credit line, so Fig. 3.11 is also similar under different thresholds. Therefore, we focus our results on a given quantile threshold $Q = 0.75$ for the remainder of the study. Although the violation of homogeneity in 3 prevents us from making a direct comparison between variables, these results imply that CI significantly and independently affects the fraction of the wealthy population.

3.4.3 Network metrics selection

Many metrics or centralities have been considered to characterize the influence or importance of nodes in a network [68, 78, 95]. Here, we consider only those centralities that can be scaled up to the large network size considered here: (a) degree centrality k_i (number of ties of individual i) is one of the simplest [68], (b) PageRank, of Google fame [42], is an eigenvector centrality that includes the importance of not only the degree, but also the nearest neighbors, (c) the k-shell index k_s of a node (Fig. 2.4a), i.e., the location of the shell obtained by iteratively pruning all nodes with degree $k \leq k_s$ [40], and (d) the collective influence of a node with degree k_i (Fig. 2.4b) in a sphere of influence of size ℓ defined by the

frontier of the influence ball $\partial\text{Ball}(i,\ell)$, and predicted to be $\text{CI} = (k_i - 1)\sum_{j\in\partial\text{Ball}(i,\ell)}(k_j - 1)$ by optimal percolation theory [43]. The detailed definition is provided in Chapt 1 and section 3.

As opposed to the other heuristic centralities, CI is derived from the theory of maximization of influence in the network [44]. The top CI nodes are thus identified as top influencers or superspreaders of information, and they do so by positioning themselves at strategic locations at the center of spheres surrounded by hubs hierarchically placed at distances ℓ (Fig. 2.4b). These collective influencers also constitute an optimal set that provides integrity to the social fabric: they are the smallest number of people that, upon leaving the network (a process mathematically known as optimal percolation [43]), would disintegrate the network into small disconnected pieces.

3.4.4 Correlation between network metrics and financial status

By definition, all of the metrics have similarities (e.g., they are proportional to k, and PageRank and CI are based on the largest eigenvalues of the adjacency and non-backtracking matrices, respectively [43]), and indeed, we find that their values in the phone communications network are correlated (Table 3.2). More interestingly, Fig. 3.11 provides evidence of correlation of the four network metrics with financial status (ranked credit limit) when we control for age, indicating that the network location correlates with financial status. In this figure, we plot the fraction of wealthy individuals (defined as top 4th quantile, equivalent to a credit limit greater than USD $4,000. See section above) for details about validation methods and [92]) in a sampling grid for a given value of age and social metric as indicated.

To compare the value of the social metrics to the economic status of individuals, we have to draw out the best one to describe network location influence effects. We sum up all the age groups and consider the effect of network metrics to demonstrate the effects of each variable.

The reason for using the aggregated model instead of the direct correlations at the individual level is because the regression models at the individual level are based on certain assumptions that are not satisfied by our data. Thus, we were unable to apply regression models at the individual level, and instead provide data at an aggregated level. The failure of regression models at the individual level is due to two reasons:

1. The distribution of credit limit (CL) for a given level of ANC [which is a log-normal-like distribution with several peaks located at integers such as 50,000 or 100,000 (Supplementary Figure 3.8a)] is not invariant under changes in ANC. That is, the distribution changes shape when ANC increases, showing an increasing fraction of high-CL population while the fraction of people around the mean value stays unchanged (Supplementary Figures 3.8b–d). Such behavior directly violates the constant variance assumption of regression models and causes the data to be poorly captured by least-square regression models.

2. Besides the above fluctuations in the credit limit, other unknown factors may provide random fluctuations in inferring individuals' financial status. Such combined random effects are considerable at the individual level. However, aggregation models reduce the fluctuation caused by random factors, and the effect of the network emerges at the population level.

Thus, we adjust our statistical model to reflect the complexity of economic effects from network metrics and aggregate the data as follows:

First we separate the individuals into groups of sampling grids in a variable space (in 1D as segment bins and in 2D as grids). In each group (with more than 10 people for statistical significance), we count the fraction of wealthy individuals defined as those individuals in the top 4-quantile $Q > 0.75$ or who have a total credit limit greater than (equivalent to)

USD $4,000. The dependence of our results on different wealth thresholds is provided in the subsection above.

Besides the degree, the volume of communication may have correlations with economic status since we could not eliminate the systematic bias caused by phone call service fees. We investigate the correlation between the fraction of wealthy people and the average communication load per link: $\mathrm{AVL}_i = \frac{W_i}{k_i}$, where W_i is the volume of communication events and k_i is the degree of node i. The regression result shown in Supplementary Figure 3.10 shows that there is no significant correlation between the average communication volume per link and the fraction of wealthy individuals. Therefore, the effect of communication volume is negligible in comparison with the other variables considered in this study.

Fig. 3.9 shows the results. The large fluctuation in degree for higher quantiles in Fig. 3.9a implies that the effect of degree involves complex social patterns rather than only the local properties of the degree of the node. Thus, we abandon the use of degree for further study as an indicator. k-shell is good enough to present a positive correlation of high network location influence. However, due to the limited values of k-core, it cannot provide finer resolution for prediction (Supplementary Figure 3.9b). Therefore, k-shell is also not considered for further studies as an indicator. The performance of PageRank (Supplementary Figure 3.9c) with a slightly negative correlation suggests that it is not the optimal variable to rank economic status, and thus it is not considered herein.

Finally, CI (Supplementary Figure 3.9d) shows strong global correlation and satisfying resolution, which makes it a convenient metric for quantifying the influence of network location. The strong correlation with CI is invariant under different radii of influence ℓ (Supplementary Figure 3.12).

We notice a non-monotonic oscillatory behavior of the fraction of wealthy people when using k and CI as variates (Supplementary Figures 3.9a and 3.9d). This effect is complex and cannot be captured by either the degree or CI, and may not be limited to local properties.

The oscillation is reduced when using CI in the analysis, and this is one of our reasons for choosing CI as a potential predictor. We will continue investigating the non-monotonic pattern in future work.

3.4.5 Composite metrics and financial status

While all of the social metrics show correlations with financial status when considered with age (Fig. 3.11), the question remains of which metric is the most efficient predictor. Strong correlations with economic wellness are observed for the feature pairs (age, k-shell) ($R^2 = 0.96$, Fig. 3.11c) and (age, CI) ($R^2 = 0.93$, Fig. 3.11d). The analysis on the section above indicating that k-shell and CI better capture the correlation with credit limit. Between these two metrics, CI guarantees a requirement for both strong correlation and sufficient resolution. K-shell cannot capture further details due to its limitation of values (k-shell ranges from 1 to 23, dividing the whole population into this small number of shells with a typical shell containing tens of millions of people), while CI spans over seven orders of magnitude (Fig. 3.6). This high resolution implies that CI is a more accurate social signature for the financial status of the individuals. According to its definition (Fig. 2.4b), a top CI node is a moderate to strong hub surrounded by other hubs hierarchically placed at distance ℓ. However, we emphasize that CI is just a useful strategy for the reasons shown above, and by no means the only or best strategy to correlate the wealth of individuals and their network influence.

While the theory behind CI is a global maximization of influence, CI represents the local approximation to this global optimization. Thus, CI represents a balance between a global optimization and its local approximation, taking into account the first 2 or 3 layers of neighbors via the parameter ℓ, which represents the size of the sphere of influence used to define the importance of a node, Fig. 2.4c. By changing ℓ, we discover that CI with $\ell = 2$ is sufficient to capture the correlation between network influence and wealth (Supplementary Figure 3.12).

To track the effect of CI independently of age we investigate the effects of CI inside two specific age groups in Figs. 3.13a and 3.13b. In both age groups, high CI is always accompanied by a higher population of wealthy people. A relatively smaller slope in age group <30 suggests that the CI network effect is more sensitive for older people with more mature and stable economic levels, than for younger people (see in Supplementary Figure 3.7). When we combine age and CI quantile ranking into an age-network composite: ANC $= \alpha$ Age + $(1 - \alpha)$ CI, with $\alpha = 0.5$, a remarkable correlation ($R^2 = 0.99$, Fig. 3.13c) is achieved. By combining network information with age, the probability to identify individuals with a high credit limit reaches $\sim 70\%$ at the highest earner level. Such a level of accuracy renders the model practical to infer individuals' financial fitness using network collective influence as we show next.

3.4.6 Network Diversity and and Financial Status

Our combined datasets also offer the possibility to test the importance of the diversity of links, as measured by ties to distant communities in the network not directly connected to an individual's own community, at the level of single individuals [69, 70, 45]. To this end, we first detect the communities in the social network by applying fast fold modularity detection algorithms which implemented as follows:

Personal structural hole [74] effects were evaluated by the ratio of total weights attached with nodes outside a community k_{out}, to those inside a community k_{in}. A fast community detection algorithm introduced by Blondel *et al.* [38] was implemented in this work. The algorithm aims to maximize the modularity function [38, 39]:

$$Q_m = \frac{1}{W} \sum_{i,j} [W_{ij} - \frac{W_i W_j}{W}] \delta(c_i, c_j), \tag{3.11}$$

where W_{ij} is the number of communication events loaded on link i, j and c_i is the community

label of node i. $W_i = \sum_{j \in \partial i} W_{i,j}$ and $W = W_{ij} \sum_{i,j}$. The global maximization of modularity was achieved by iteratively calculating the local maximization of normalized networks based on communities. Different communities were labeled during each iteration. Among all the communities, we chose the clustering of the second iteration to control the average scale of the community to 10^2. There are 4.92×10^5 communities inside the network. The distribution of community sizes is fat-tailed with a largest community size of 10^6 (Supplementary Figure 3.14). The fraction of wealthy individuals inside each community is independent of the size of the community ($r < 0.05$).

After we label the network with its communities, we can evaluate an individual's structural hole effect [74] by introducing the diversity ratio DR. DR is defined by the ratio of total communication events with people outside one's own community W_{out} to those with people inside the community, namely W_{in}, DR $= W_{\text{out}}/W_{\text{in}}$. The ratio is weakly correlated with CI ($r = 0.4$). The same statistic of composite ranking was implemented as CI with the same number of statistic segments and composite factor $\alpha = 0.5$ as in the text. The result (Fig. 3.13d) shows that the structural hole effect also has a strong correlation with the distribution of affluent individuals while it is weakly dependent on CI. This result confirms the importance of the ability to communicate with outside communities via "weak ties" for personal economic development [70].

The diversity of an individual's links can be quantified through the diversity ratio DR $= W_{\text{out}}/W_{\text{in}}$ [74], defined as the ratio of total communication events with people outside their own community, W_{out}, to those inside their own community, W_{in}. This ratio is weakly correlated to CI ($R = 0.4$), suggesting that it captures a different feature of network influence. We implement the same statistics of composite ranking as before, resulting in an age-diversity-composite ADC $= \alpha$ Age $+ (1 - \alpha)$ DR, with weight $\alpha = 0.5$. The result (Fig. 3.13d) shows that ADC correlates with individual financial wellbeing, generalizing the aggregated results in [45] to the individual level. Thus, the social metrics considered,

DR and CI, express the fact that higher economic levels are correlated with the abilities to communicate with individuals outside one's local tightly-knit social community, a measure of Granovetter's "strength of weak ties" principle [69] and to position oneself at particular network locations of high CI that are optimal for information spreading and structural stability of the social network. We note that no causal inference can be established with the present data.

3.4.7 Validation by Marketing Campaign

To validate our strategy we perform a social marketing campaign whose objective is the acquisition of new credit card clients, by sending messages to affluent individuals (as identified by their CI values) and inviting the recipients to initiate a product request. In the text we sent during the campaign, we did not provide a specific product. Instead, the only information we provided was to notify the client that he/she was eligible for an offer from the bank. This somehow eliminated the bias caused by the nature of a product which may have a different appeal to wealthy or poor people.

We note that in this experiment we use an independent dataset from a different time frame, and we use only the CI values extracted from the network to classify the targeted people. Specifically, we use the communications network resulting from the aggregation of calls and SMS exchanged between users over a period of 91 days. The resulting social network contains 7.19×10^7 people and 3.51×10^8 links. The campaign was conducted on a total of 656,944 people who were targeted by an SMS message offering the product according to their CI values in the social network. We also sent messages to a control group of 48,000 people, chosen randomly. To evaluate the campaign, we measured the response rate, i.e., the number of recipients who requested the product divided by the number of targeted people, as a function of CI. In the control group, the response rate to the messages was 0.331%. Our results show that groups of increasing CI show an increase in their response rate, with

a sound three-fold gain in the rate of response of the top influencers (as identified by top CI values) when compared to the random case. When we compared the response of the high CI to the lowest CI people, the response rate increased five-fold. The results of the experiment are summarized in Table 3.4 and in Fig. 3.15.

3.5 Discussions and conclusions

This result highlights the possibility of predicting both financial status and benefits of socially-targeted policies based on network metrics, leading to tangible improvements in social marketing campaigns. The high performance of CI among network metrics also suggests the possible role of accessing and mediating information in financial opportunity and wellbeing [70]. This has an immediate impact in designing optimal marketing campaigns by identifying the affluent targets based on their influential position in a social network. This finding may be also raised to the level of a principle, which would explain the emergence of the phenomenon of collective influence itself as the result of the local optimization of socio-economic interactions.

Table 3.1: Results of the group entropy analysis for the wealthy and poor population.

wealthy population: quantile ranking $Q > 0.95$, poor population: quantile ranking$0.05 < Q < 0.1$

		S	$R_{\mathrm{c}}(\ell)$	$R_{\mathrm{n}}(\ell)$
$\ell = 1$	wealthy	6.37±0.12	5.5±0.4	9.3±0.7
	poor	6.68±0.10	4.3±0.3	7.1±0.5
$\ell = 2$	wealthy	7.94±0.10	141.3±4.7	$6.3 \pm 0.2 \times 10^2$
	poor	8.38±0.14	101.6±3.4	$3.1 \pm 0.1 \times 10^2$
$\ell = 3$	wealthy	9.11±0.11	443.0±11.5	$7.6 \pm 0.4 \times 10^3$
	poor	9.30±0.12	390.9±6.0	$4.9 \pm 0.4 \times 10^3$
$\ell = 4$	wealthy	10.23±0.02	565.4±10.7	$5.10 \pm 0.04 \times 10^4$
	poor	10.23±0.04	517.0±9.0	$4.23 \pm 0.05 \times 10^4$

Table 3.2: Correlation (r-values) between the metric centralities obtained from the social network and age.

	k	k-shell	PageRank	$\log_{10}$CI
Age	-0.021	-0.016	-0.033	-0.007
k		0.972	0.648	0.953
k-shell			0.589	0.960
PageRank				0.575

The correlation between gender and other features is presented through the Point-Biserial correlation coefficient, and other correlations are Pearson correlations. Point-Biserial correlation coefficients quantify the male as 1 and female as 0 and are defined as: $r = \frac{\bar{X}_1 - \bar{X}_0}{s_{n-1}} \sqrt{\frac{n_1 n_0}{n(n-1)}}$. n is the total number of samples. n_1 and n_0 refer to the population inside each group. $\bar{X}_1$ and $\bar{X}_0$ are the means of the variables in each group. s_{n-1} is the estimated unbiased standard deviation of X: $s_{n-1} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2}$.

Table 3.3: Correlation between covariate CI and independent variables: age, gender and Index of Community Wealth (ICW).

	CI	Gender	ICW
Gender	-0.0419		
ICW	-0.0093	0.0131	
Age	-0.0007	-0.0116	-0.0022

Table 3.4: Results of the real-life marketing campaign.

CI range	Count	Quantile	Answered Yes	Response Rate
[0,48]	66495	0.1	170	0.26%
(48, 246]	65164	0.2	218	0.33%
(246, 600]	65961	0.3	316	0.48%
(600, 1144]	65376	0.4	332	0.51%
(1144, 1992]	65477	0.5	363	0.55%
(1992, 3408]	65477	0.6	458	0.70%
(3408, 6032]	65736	0.7	493	0.75%
(6032, 11772]	65641	0.8	555	0.8%
(11772, 28740]	65683	0.9	657	1.0%
(28740, 2719354]	65683	1.0	573	0.87%

Individuals ("Count") were targeted according to their quantile CI ranking in the whole social network obtained from phone communications activity. The response to the campaign ("Answered Yes") was computed to calculate the Response Rate.

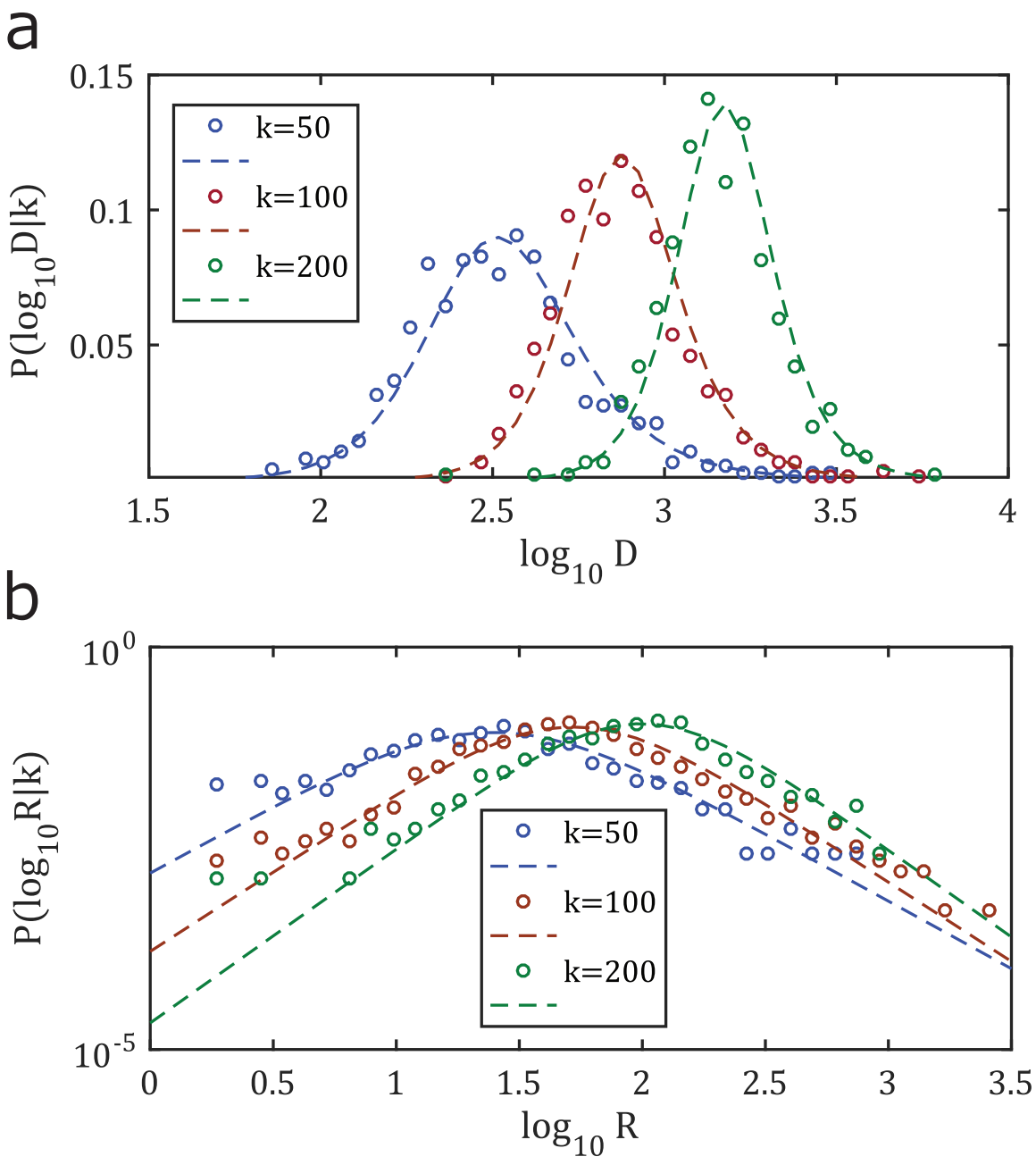

Figure 3.1: Logistic fitting result for $k = 50$, 100 and 200.
The result of paired communication R is presented in log-log scale in order to highlight the fitting for the exponential tails.

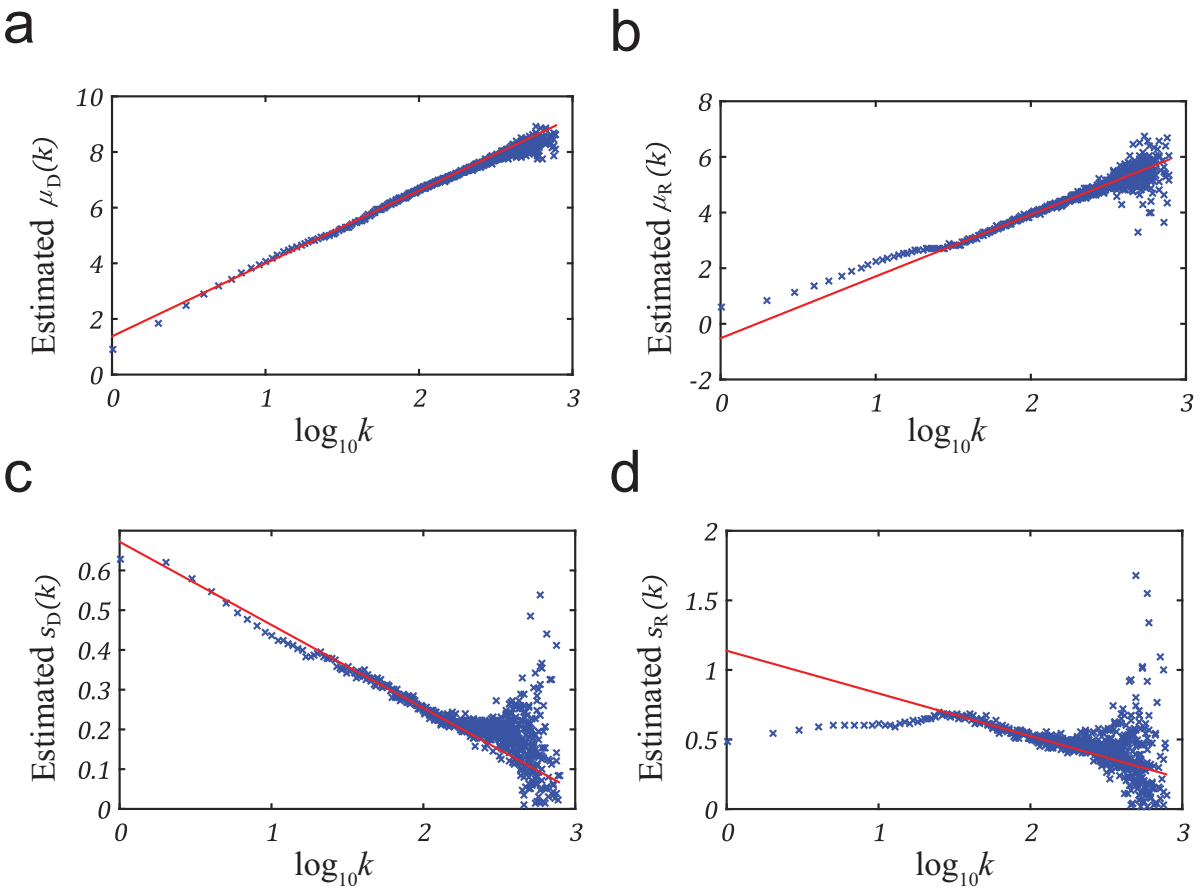

Figure 3.2: Scaled parameter estimation and its linear fitting: (a) $\hat{\mu}_{\mathrm{D}}(k)$, (b) $\hat{s}_{\mathrm{D}}(k)$, (c) $\hat{\mu}_{\mathrm{R}}(k)$, (d) $\hat{s}_{\mathrm{R}}(k)$.

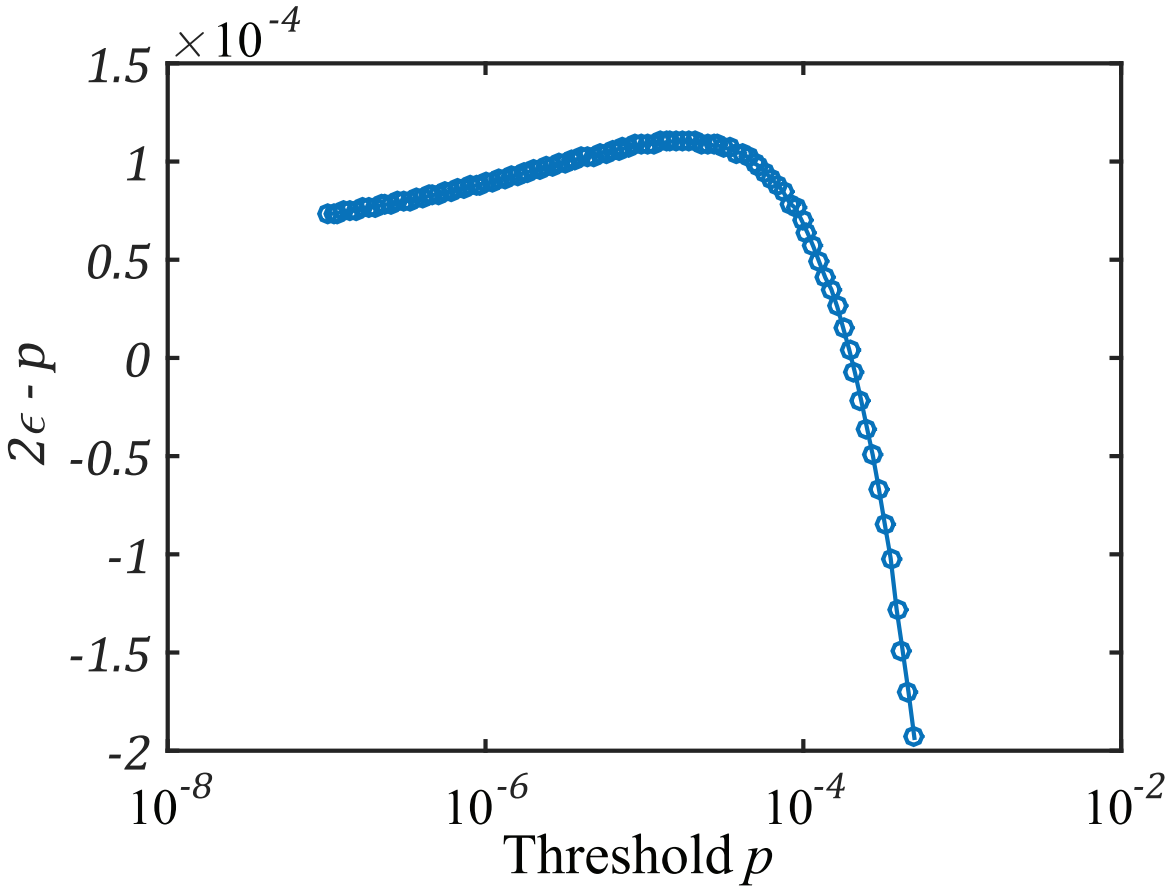

Figure 3.3: Number of outliers $\epsilon - 2p$ vs cut-off threshold p. Maximum is reached when $p \sim 1.6 \times 10^{-5}$.

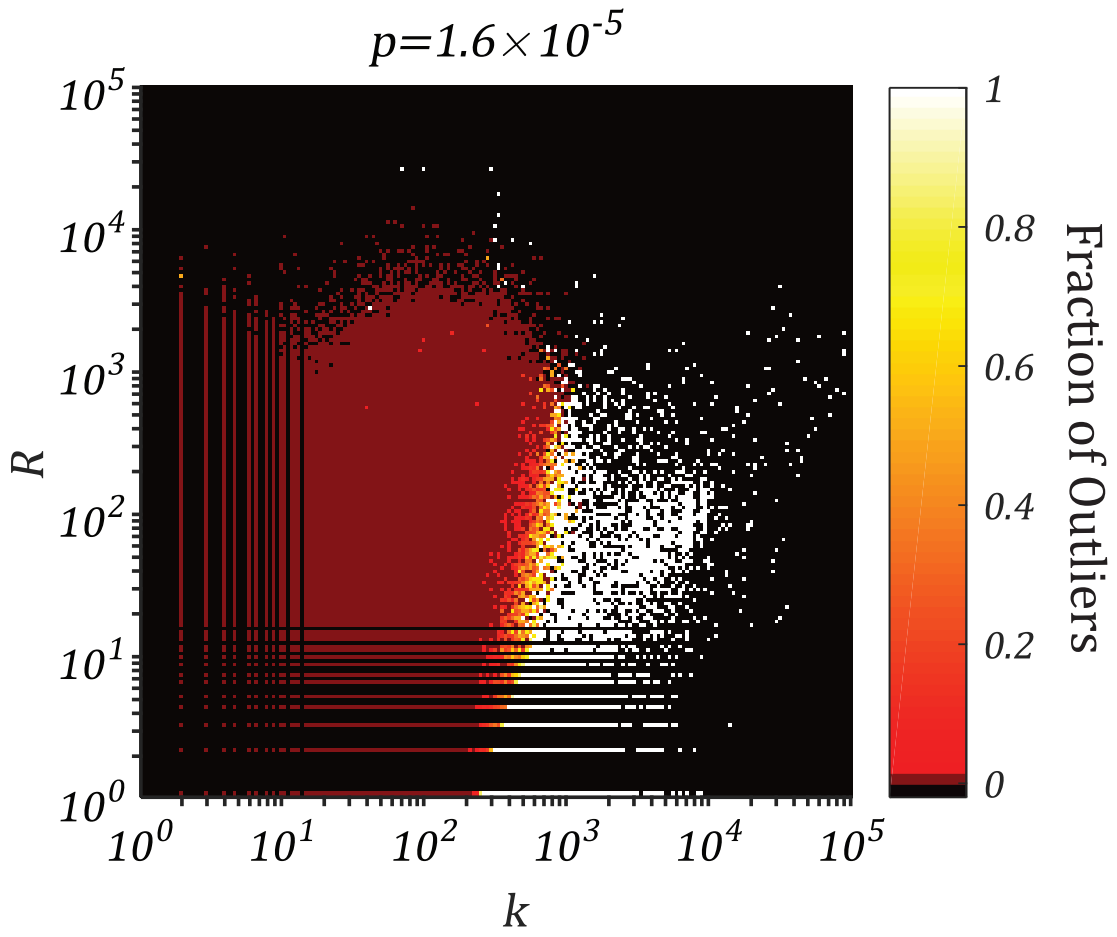

Figure 3.4: Final result of data filtering.
The result is presented in the space of k and communication pairs R. The data points were put into a grid bin of 200×200. The color represents the fraction of outliers in each bin. The filter gives us a gradual boundary of human- and non-human-operated lines.

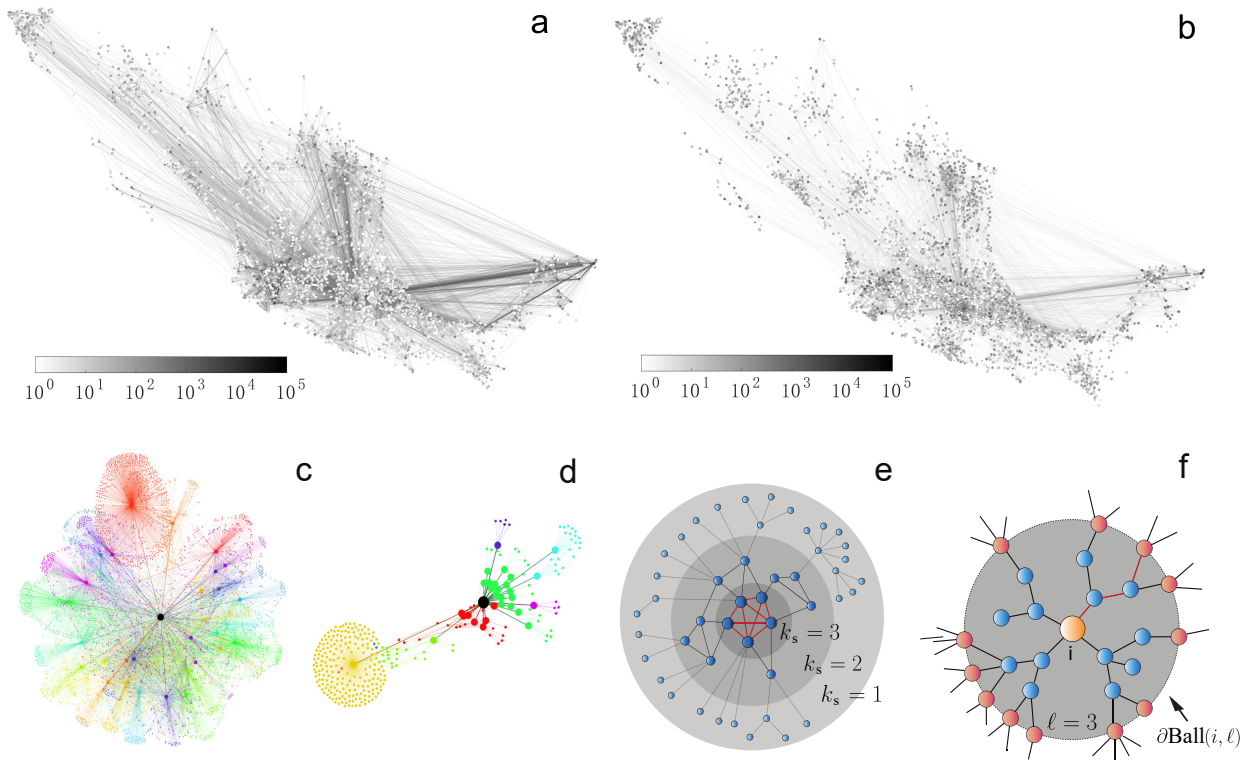

Figure 3.5: Patterns of network influence mimic patterns of income inequality Visualization of communication activity of the population in **(a)** the top 1% (with credit limit larger than USD $25,000, converted, in the country of study) and **(b)** bottom 10% (with credit limit smaller than USD $600, converted) of total credit limit classes. Links are between bank clients who have registered their zip code. Resolution of both plots is 1700×1000. The number of bank clients inside each community is reflected by the size of the node. Average credit limit is denoted by a node's grayscale. The color and thickness of the edges reflects the number of communication events between different communities. **(c)** Examples of the ego-network (extended to two layers) for an individual in the top 1% wealthy class and **(d)** an individual in the bottom 10% class. The networks show two distinct patterns of social ties according to high and low economic status: the former is characterized by large CI, the latter by low CI.

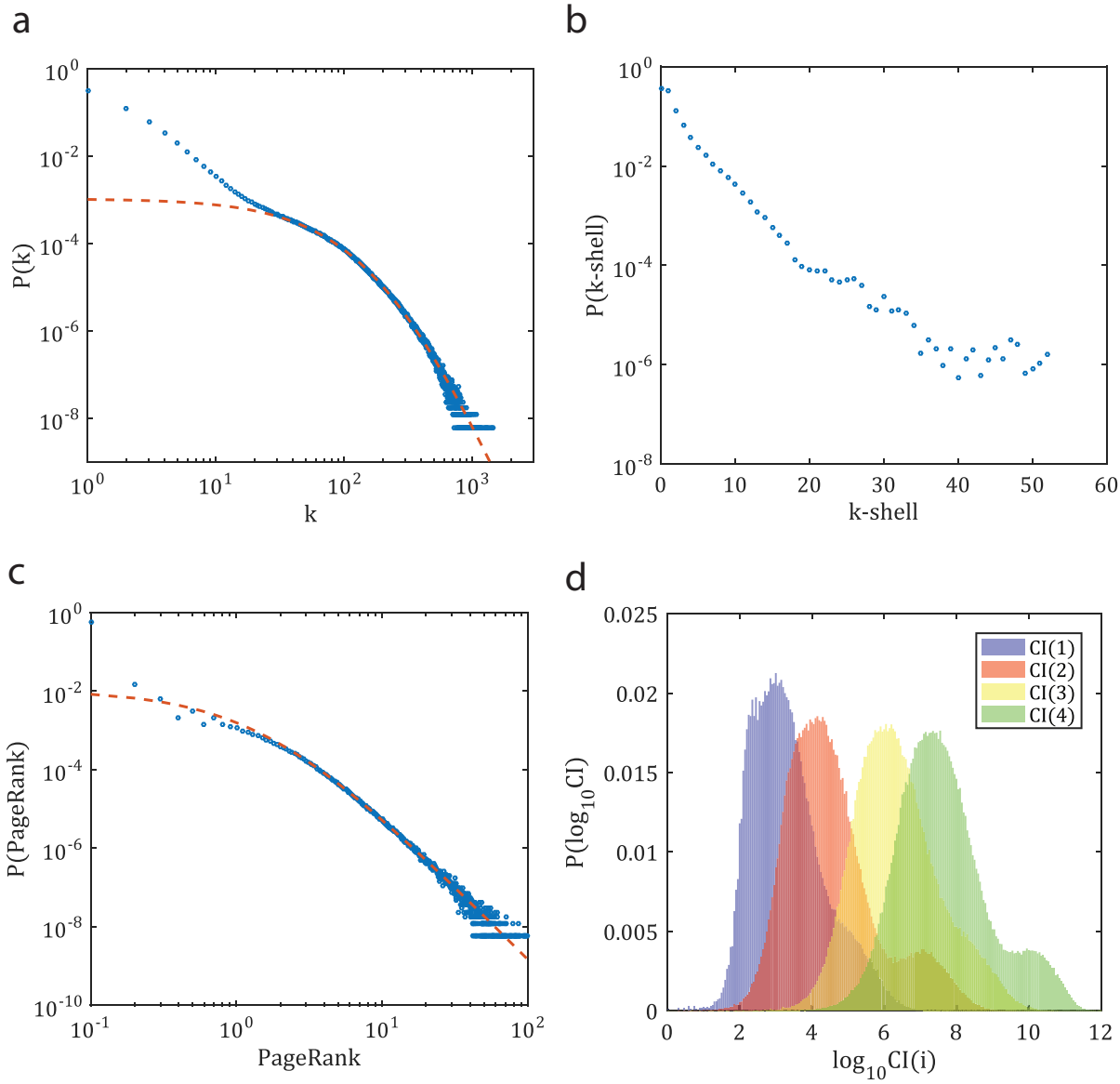

Figure 3.6: Distribution of network metrics.
(a) degree, **(b)** k-core, **(c)** PageRank, and **(d)** Collective Influence (ℓ =1 to 4). Collective Influence follows a double-tailed distribution. A small peak for larger CI emerges for even ℓ.

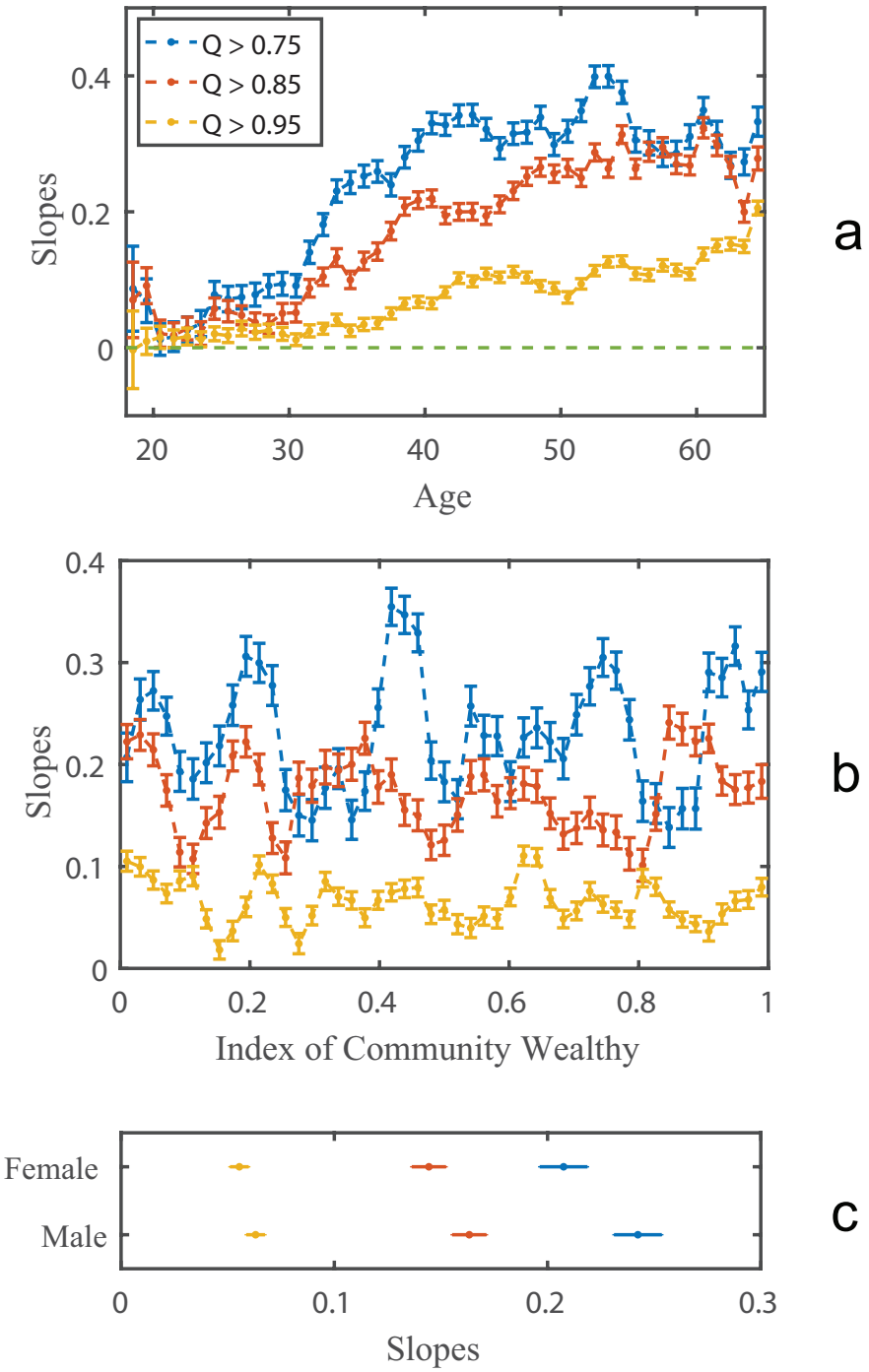

Figure 3.7: Estimated slopes in different groups of independent variables. (a), Age, (b), Index of Community Wealth (ICW), and (c), Gender. 95% confidence interval is marked by error bars in the plot. Different thresholds of wealth Q are labeled by different colors.

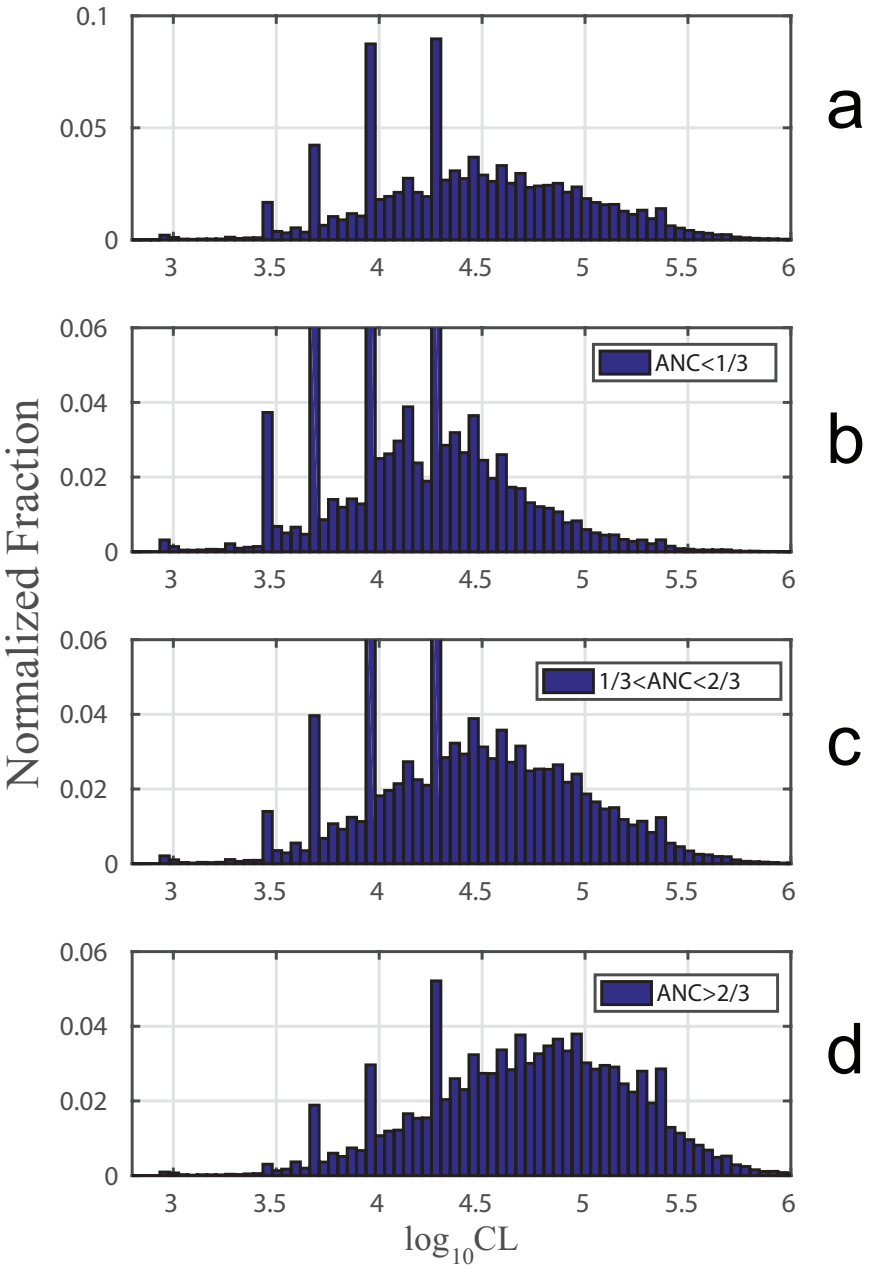

Figure 3.8: Distribution of Credit Limit (CL) under different age-network composite (ANC) groups.

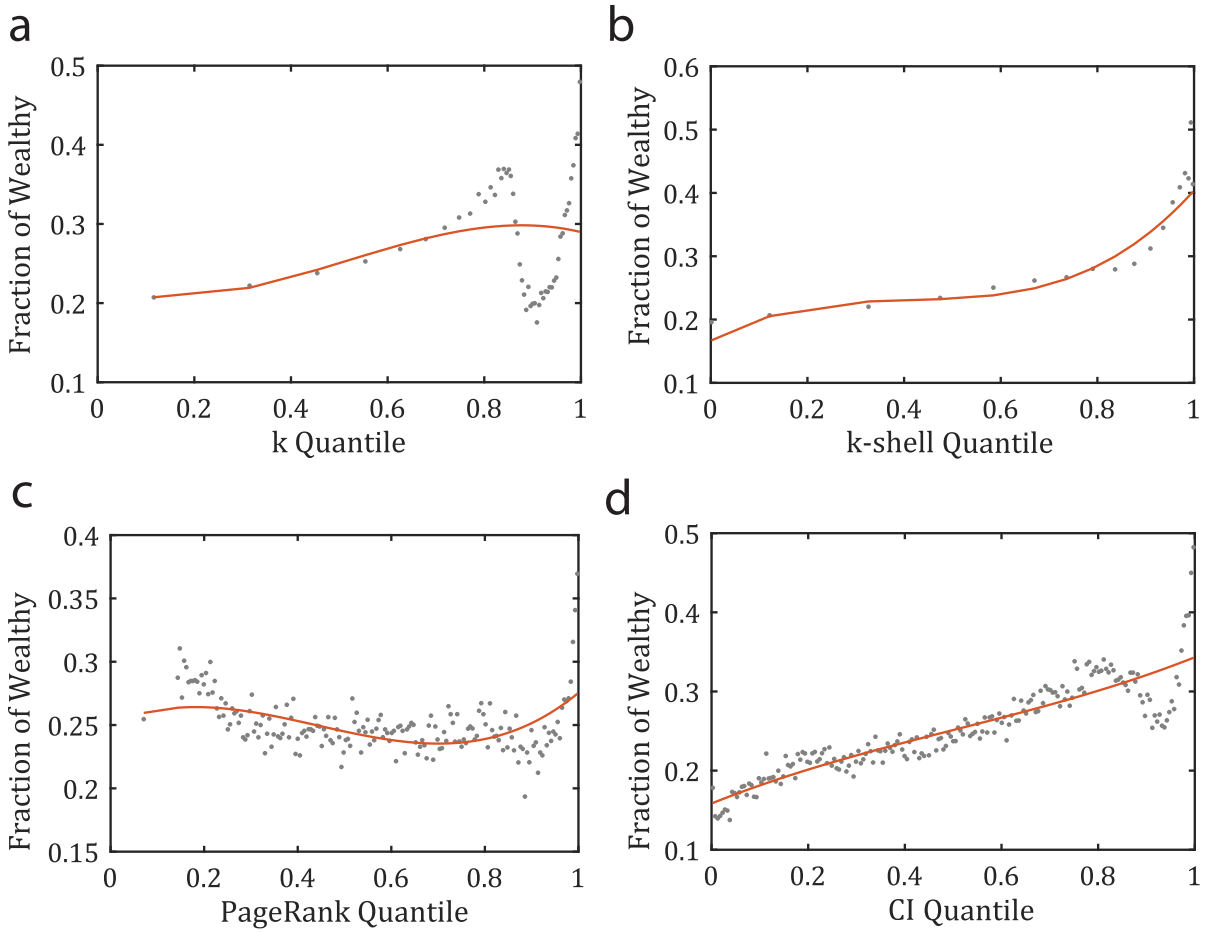

Figure 3.9: Fitting results of wealthy population vs. network influence metrics along with corresponding R^2 values.

(a) Degree (0.51), **(b)** k-core (0.99), **(c)** PageRank (0.28), and **(d)** Collective Influence (0.80). All variables are normalized to [0, 1] by the quantile ranking to ensure an adequate number of data points in each partition. The entire quantile ranking is divided into 200 segments from minimum to maximum. Only those groups with population larger than 10 are shown on the plot. Out of the four metrics, CI is the most convenient for capturing high correlations and presenting a large range of values that allow us to classify the whole population.

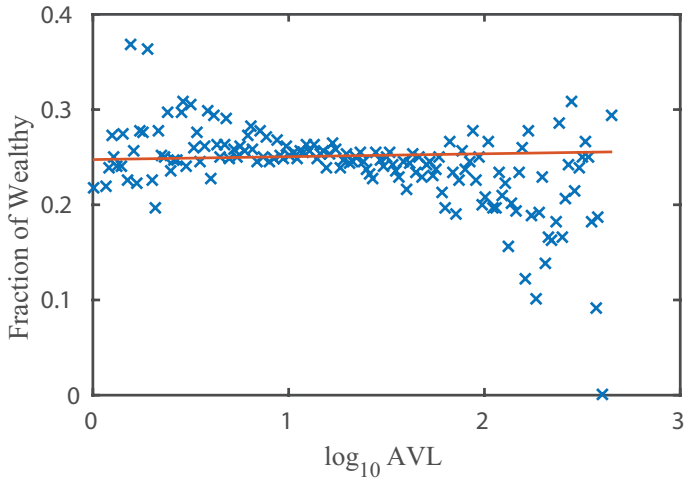

Figure 3.10: Fraction of wealthy people vs. average communication event load per link (AVL).

AVL is in log-10 scale and divided into 200 partitions. Each group with a population of more than 10 is considered in counting the fraction of wealthy people inside the group.

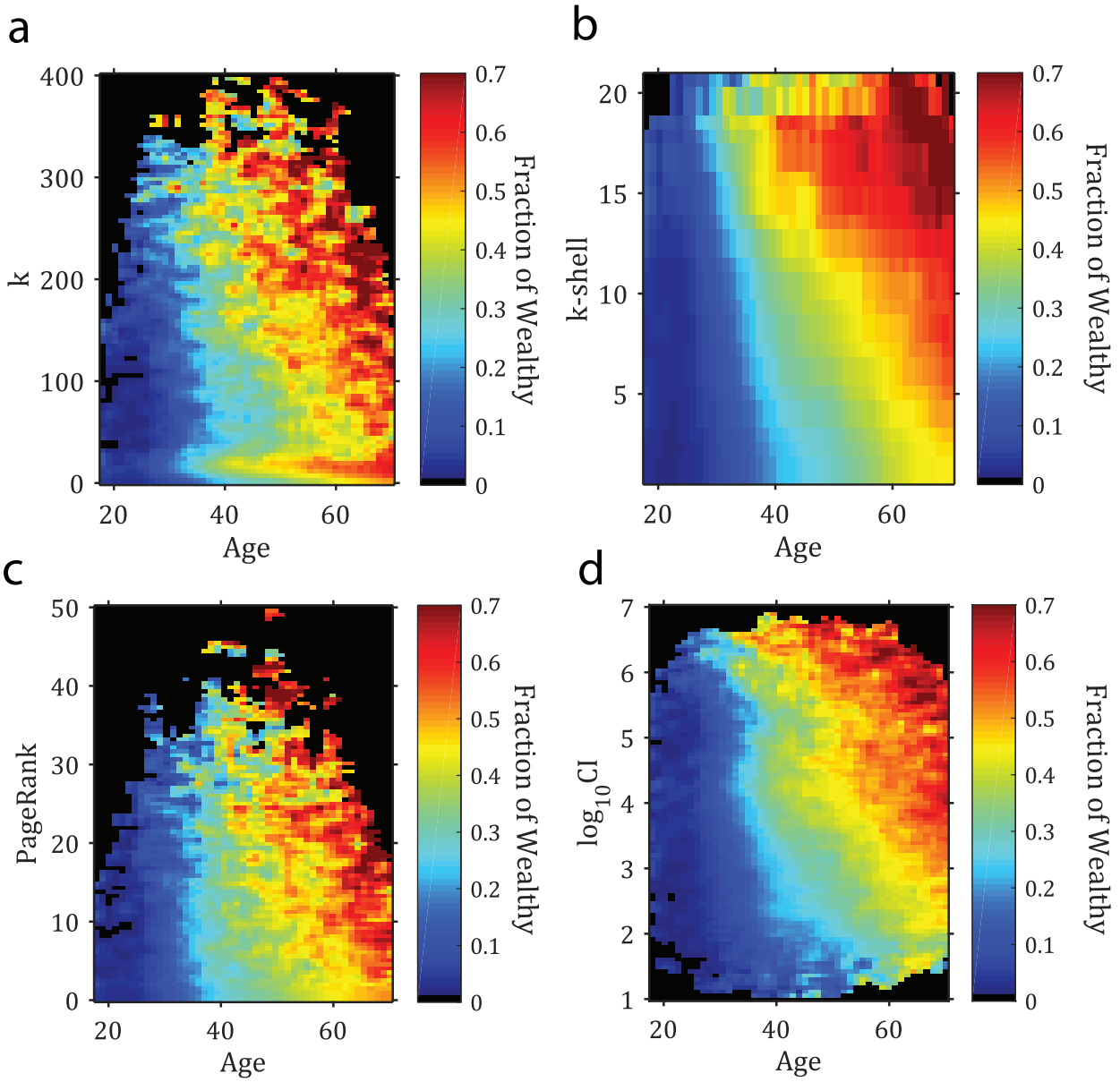

Figure 3.11: Fraction of wealthy individuals vs. age and network metrics. Correlation between the fraction of wealthy individuals vs. age and **(a)** degree k ($R^2 = 0.92$), **(b)** k-shell ($R^2 = 0.96$), **(c)** PageRank ($R^2 = 0.96$), and **(d)** $\log_{10}$CI ($R^2 = 0.93$). Only those groups with population larger than 20 are shown in the plot. The four metrics correlate well with financial status when considered with age. Further correlations studies indicating that CI could be considered as the most convenient metric out of the four due to its high resolution.

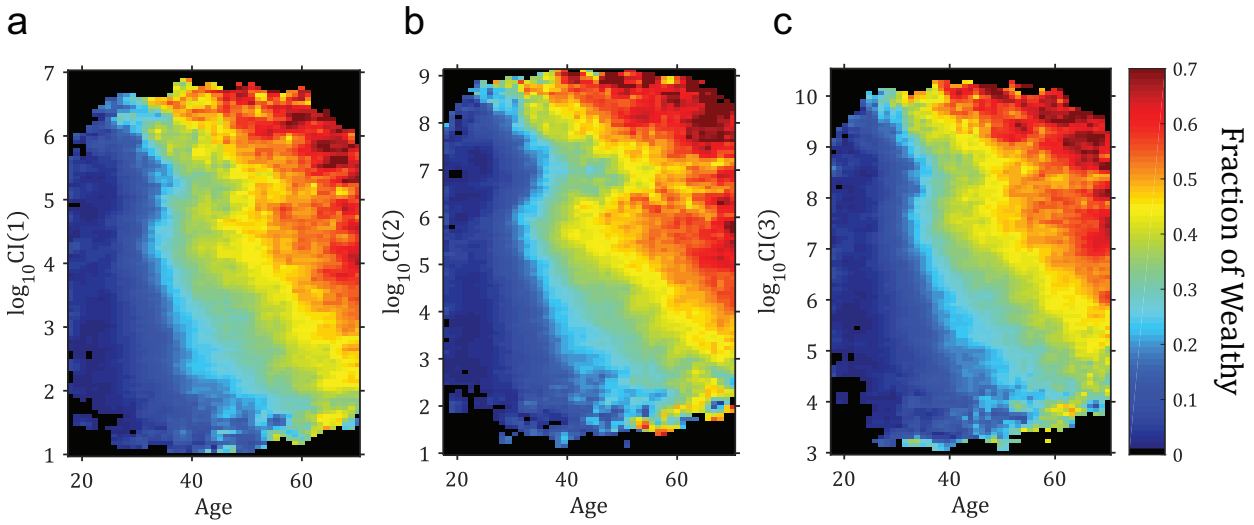

Figure 3.12: Fraction of wealthy people in each group against age and logarithm collective influence for different radius.
Radii ℓ range from 1 to 3. Communities are determined by 200 segments covering from the bottom 1% to top 1% of CI values. Only those groups with population larger than 10 are shown on the plot.

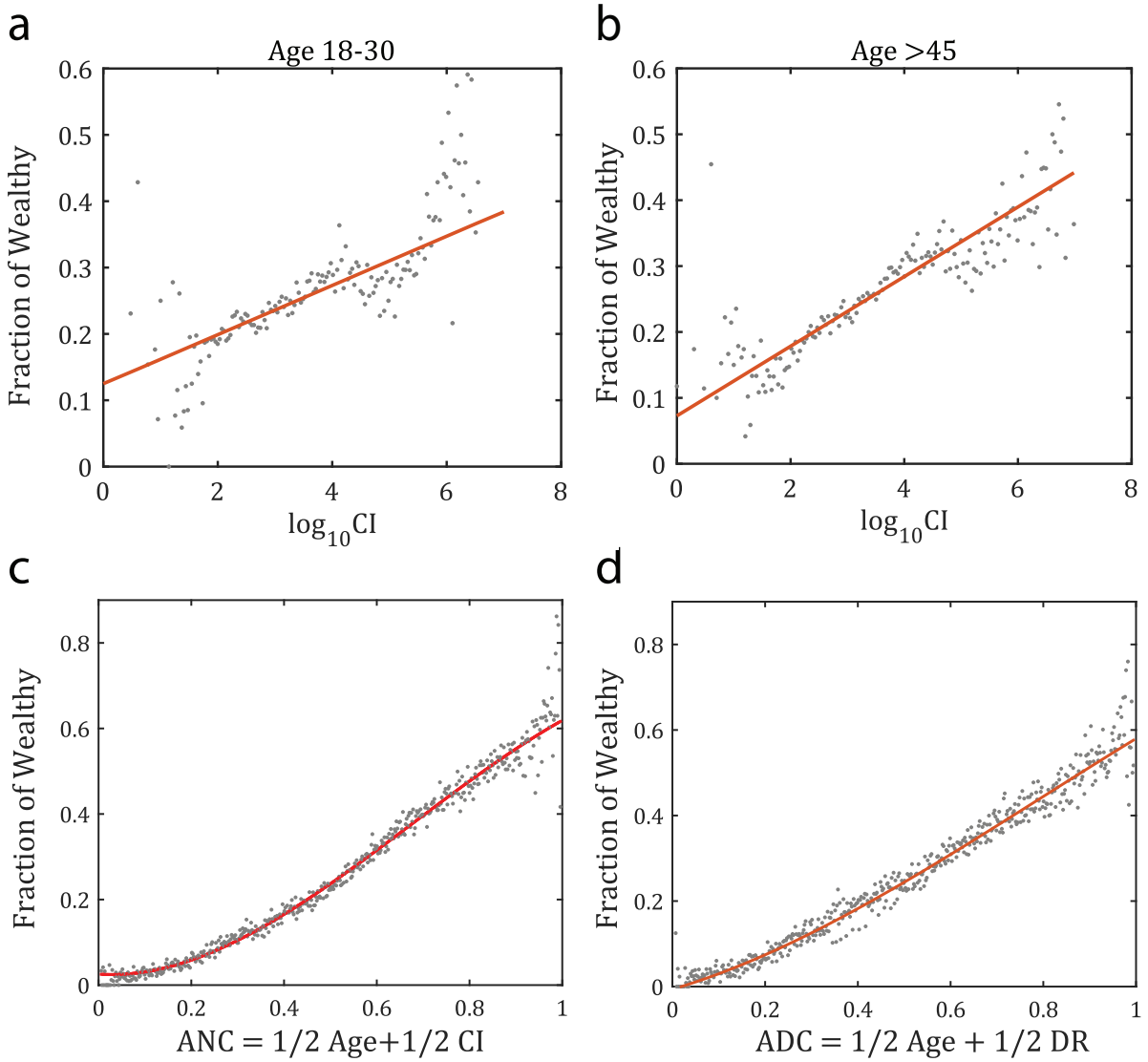

Figure 3.13: Fraction of wealthy individuals over different age and composite ranking groups. Correlation between the fraction of wealthy individuals as given by the top 25% credit limit and CI in different age groups of **(a)** 18-30, **(b)** >45. Correlations between top economic status and large collective influence as determined by CI values in different ages are significant in all age groups, while the slope of the linear regression is larger in the older group (0.053 compared to 0.037). **(c)** Age-network composite ranking ANC = 1/2 Age + 1/2 CI, and **(d)** age-diversity composite ranking ADC = 1/2 Age+ 1/2 DR. By combining the network metrics with age into a composite index, the chance to identify people of high financial status reaches $\sim 70\%$ for high values of the composite. Both R^2 show a high level of correlation ($R^2 = 0.99$ and 0.96 for ANC and ADC, respectively), making both composites good predictors of wealth in practical applications.

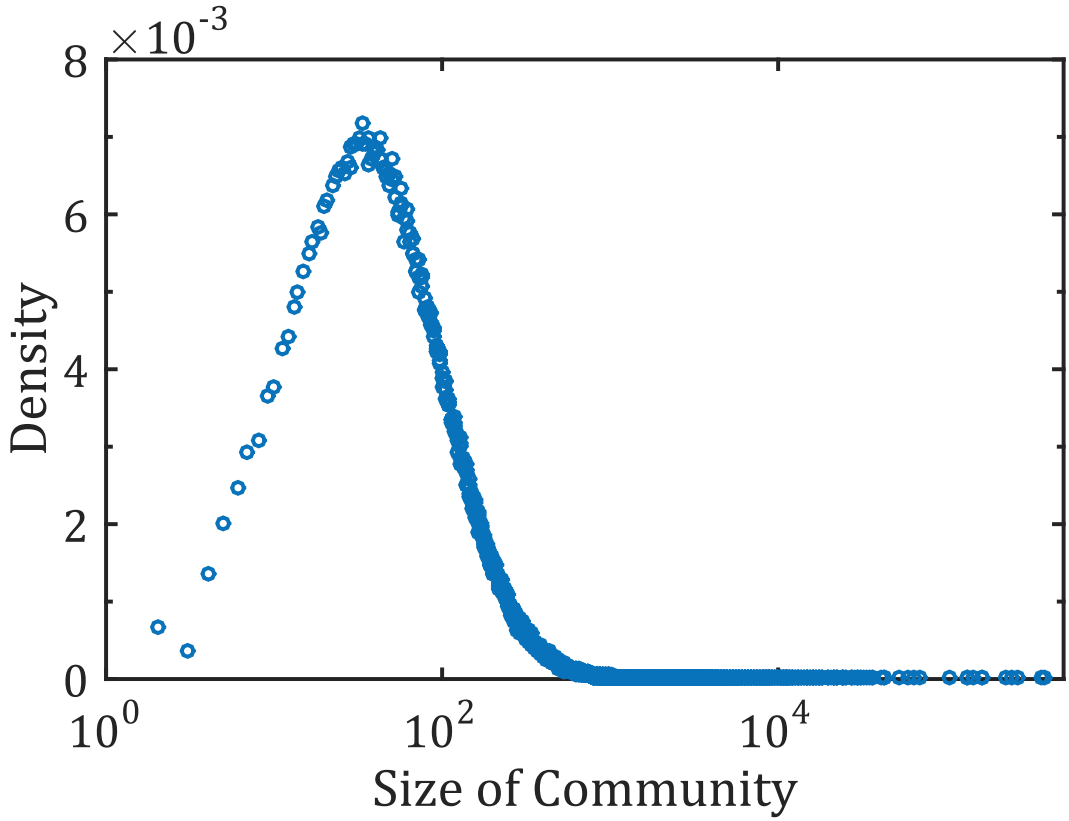

Figure 3.14: Distribution of community sizes in the entire social network at second iteration.

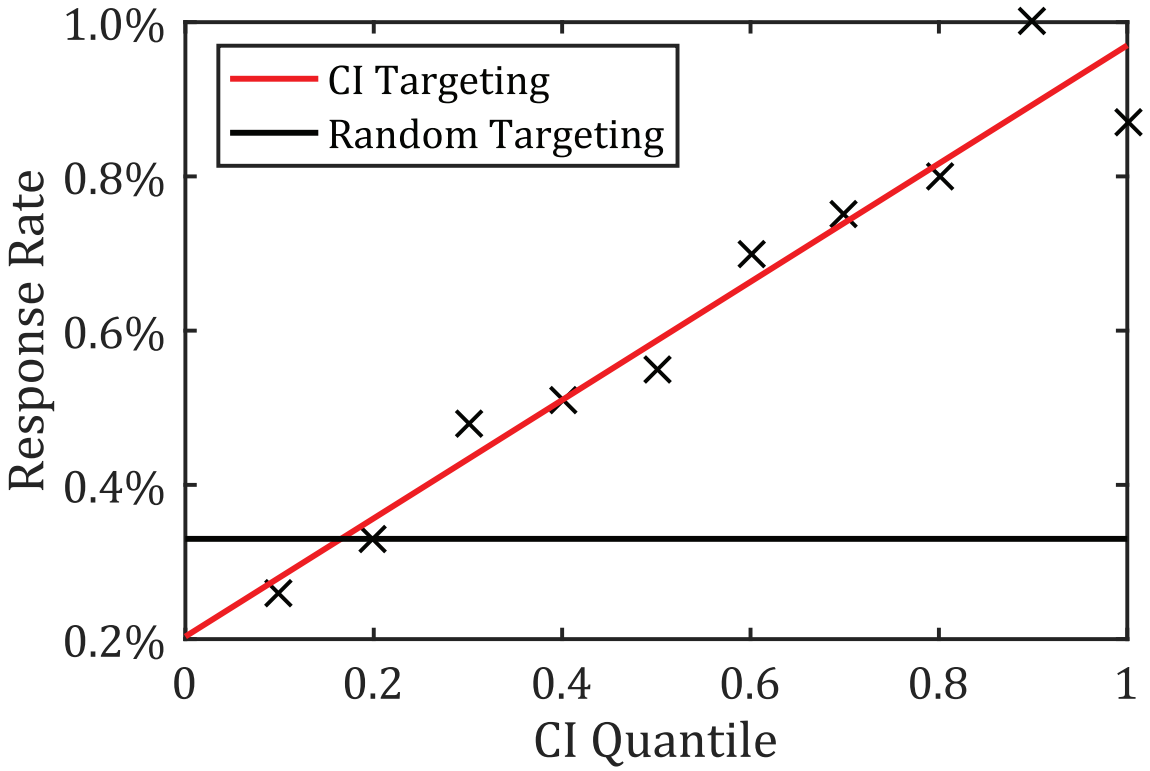

Figure 3.15: Response rate vs. CI quantile in the real-life CI-targeted marketing campaign. The response rate increases approximately linearly with CI ranking. The CI-targeted campaign shows a three-fold gain for the top influencers with high CI, as compared with a campaign targeting a randomized control group.

Chapter 4

Application 2: Deep Learning in Prognosis of Breast Cancer

In this chapter, we are introducing another application of network models in machine learning: Network-based deep learning models in predicting pathology result of breast tumor via MRI scan images. The following sections will demonstrate how the combination of deep-layer network models contribute to a real world complex classification problems. The behavior of such complex model will be investigated and evaluated by a standard testing procedures.

4.1 Problem Description

Breast cancer is a cancer developed from breast tissue. It has become the most major thread on women's life Worldwide, this invasive cancer affects about 12% of women worldwide and comprises 22.9% of invasive cancers in women [96]. Most early-stage breast cancer can be controlled or even cured by continuously and proper treatment [97]. Due to the high occurrence and fatalness of the cancer, oncologist suggest the population with high-risk population to take regular body examination to detect the sign of cancer earlier.

By the research so far, risk factors for developing breast cancer include: being female, obesity, lack of physical exercise, drinking alcohol, hormone replacement therapy during menopause, ionizing radiation, early age at first menstruation, having children late or not at all, older age [98]. Another significant factor is the inheritance of risky genes from family. About 510% of cases are due to genes inherited from a person's parents [99], including BRCA1 and BRCA2 among others. Genetic testing is the most efficient way in filtering these population[100]. In American, many hospitals provide genetic test for patiences to determine whether they have risky genes for breast cancer and ask for a regular examination of breast.

The signs of early breast cancer including physical change of breast tissue, cysts and unidentified tumor (mostly are benign) [97]. The detection methods can varies from physical examination, ultrasound, micrograph (Fig. 4.1a), Magnetic Resonance Imaging (MRI) (Fig. 4.1b) and biopsy[101]. The biopsy is the most accurate method which directly take tissues from breast. However, this method requires that the pathologist know exactly the location of suspicious tissue and usually it takes time to obtain the result and is painful to the patients. Micrograph and MRI imaging will quickly locate the suspicious areas and provide preliminary visual information (like the size, shape, texture of a tumor). Nevertheless, the disadvantage of micrograph and MRI imaging obvious: the numbers of machine operating the test is limited and the cost is high for most patients.

Due to the large amount of risky gene carriers and limited medical resources, it is unlikely to make every risky person to do the expensive MRI scans in a very frequent pace (like once half an year). In the following sections, we discover that in most cases even a risky patient with benign tumor in the breast, the chances of this tumor transforming into malignant is still low if proper treatment is received. Therefore, for most patients, the frequency of the MRI scan can be reduce to once or year or even less. However, by the means so far, pathologists make the empirical diagnose based on her/his own experiences and some basic

patterns. Such decisions may varies from doctors to doctors and the intensive examine on daily basis will increase the error rate made by pathologists.

Recently, artificial intelligence (AI) equipped with image recognition models has been widely used in oncology diagnosis. Some models has been developed based on Convolutional Neural Networks (CNN) in classifying skin cancer ([102]), finding the knots in lung ([103]) and brain tumor segmentation [104]. However, the application of AI on MRI images are limited especially for breast cancer. Classical feature methods are developed [105] to assist the pathologists to grade the type of tumor and breast and fibroglandular tissue [106]. The state-of-the-art application of deep learning in breast cancer is to detecting the area of malignant tissues via micorgraph [107]. The accuracy of such AI methods are promising and comparable to pathologists. AI's has advantages in recognizing the anomaly because usually they trained on a huge amount of samples. Another important reason is AI always record tiny features that usually beyond recognition of human so that the decision will be made based on more details even though it is hard to interpreted. However, AI also have limitations. First the AI are learning based on statistical model which requires a large amount of training set. Second, most AI models are designed for particular cases. Last but not the least, the classification process of AI is very difficult to interpreted which make it very difficult to find information from the model. Especially, there are very few models developed base on MRI images due to the following reason:

1. MRI scan are 3-D space imaging technique which results in the output is a 3-D tensor while most of the image recognition model are based on 2-D images.

2. To apply a 3-D CNN on MRI images are not practical. This is mainly because the number of parameters on 3-D CNN are large even with a simple model but usually the number of MRI images for training are limited.

3. The topology of the class distribution will be more complicated if in a 3-D space. It

requires a fine-tuned model design to reflect the complexity.

4. Information is sparse in 3-D MRI scan space. Usually, suspicious tissues only occurs in a small area while the rest of breast are normal. Area filtering are necessary before proceed to further analysis.

Nevertheless, most of the obstacles can be overcome by adequate MRI data and appropriate priors. Good priors may include: a matured pre-trained models that is used for the general classification problem, well labeled images indicating the location of suspicious tissues and some reference image from points of different time series.

Combined with all the informations we can access. We would like to build a working frame based on the deep learning technique to develop a tool to assist pathologists in diagnosing pathology result of a breast tumor. The purpose of the tool can be separated into the following stages:

Stage 1 Picking up the area in which is likely to be or containing a tumor.

Stage 2 Predict the pathology result of tumor (benign or malignant)

Stage 3 For a patient, given her/his past scan history, predicting the tumor growth in the future.

This three stages are considerably important in increasing the efficiency of pathologist in filtering the real risky cases. It will largely save the time and work which will result in a more efficient distribution of medical resources to wherever is urgently needed.

4.2 Data Description

We conduct a project collaborate with Memorial Sloan Kettering Cancer Center (MSKCC), one of the leading cancer research center in the world. MSKCC provides us almost anonymized 250,000 breast MRI images during the period from 2001 to 2016. These 250,000 images belong to 76,795 scans on 22,809 breasts cases for about 12,000 subjects. The types of scans

are determined by the signal sequences while making the images. There are three main type of MRI sequences:

1. T1: The timing of radio frequency pulse sequences used to make T1 images results in images which highlight fat tissue within the body [108]. Usually, radiologists will performing a fat saturation operation to reduce the brightness of fat tissue in order to show the detail of the tissue inside.

2. T1 with contrast: T1 with contrast use the same imaging sequence as T1 without contrast. The only difference is before the scan, the subject will be injected with some contract agent (usually is Gadolinium based) to control the signal decays of different tissue. This kind of sequence will result in the highlight of desired tissue. Usually, T1 with contrast enhanced MRI will be a series of images as a function of time.

3. T2: Unlike T1. T2 images highlight both fat tissue and water within the body. T2 sequence will indicate the bio-activity of specific tissues because an activate tissue has a faster pace of metabolism which leads a larger amount of water consumption. [108]

Once a scan is made by MRI machine, images with the above three different sequences will be automatically generated. Hence a patient may have multiple images per scan. These images are identical in outline shapes except for the contrast and highlights. For each scan, the radiologist usually scans the both sides of breasts so that for each subject, they will create 1 or 2 breast scans each time they came to hospital and received MRI screening. Some cases (13,167 out of 22,809) have at least one follow up scan after first time they got scanned. The date and time for the follow up scans are also recorded in the original file so that we can track the cases. In our work, we trained and tested our model on T1 contrast and T2 sequences because these two sequences are believed by pathologists to be the best to recognize the nature of tumors.

Despite the original anonymized images, we also obtained the anonymized clinical results from pathologists as the label of each scan. The scans are labeled as a score called BIRADS which ranges from 0 to 6. BIRADS are the scores assigned by the pathologists to reflect the risk of malignance of the breast according to the MRI scan. Scans with $BIRADS \leq 3$ usually consider as non-risky scans no matter whether there are tumors inside. Therefore, the scans with $BIRADS \leq 3$ (9,078 out of 76,795) will not send to biopsy for further examination. The pathology result is automatically labeled as B̈enign.̈ The rest of the scans are $BIRADS \geq 4$ which is consider by pathologists as dangers cases. In this scenario, further examination of biopsy will be taken to determine whether the tumors are benign or malignant. Among all the 9,078 scans with $BIRADS \geq 4$, there are 4,625 (50.9%) of them are malignant.

We notice that less than half of the cases sent to biopsy are true malignant. In the task 1 and 2, we are aiming to developing a tool to assist doctors to reduce the number of cases send to the biopsy while maintaining the sensitivity to detect all true malignant result. And for task 3, we want predict the pathology result through the follow up cases to see whether our deep learning algorithm can detect the early sign of transformation before human pathologist.

4.3 System Design

Fig. 4.2 shows the design of the system. The system is consist of three parts: feature extraction agent, image selection agent and pathology prediction agent. The key of the system idea is to reduce the 3-D image recognition problem into a combination of 2-D image recognition problem. The idea is to mimic the pathologists when they infer the nature of the tumor. 3-D MRI images was divided along the sagittal axis (y-axis which is the vertical plains generated from the screening left to right) into several image slices. Because the size

of the MRI images may vary from scans to scans, the number of result slices also varies.

The three agents worked as a combined workflow to evaluate the pathology results based on the image slices selected. Feature extraction agent first extracting the features from the images to reduce the dimensions of the system. Then the features of each slice are send to the slice selection agent to select the slices where the tumors most likely to locate. After that, the selected slices will be send to the pathology prediction agent in which the score of malignance of each slice will be evaluated and then combined as the final output indicating the prediction result.

Due to the deep structure of this framework, number of parameters will be considerable if we did not apply any priors. Nevertheless, by carefully design the training and inferring process and using a pre-trained network are the possible ways to avoid over-fitting of the system. In the sub-sections below, we will describe each agent in details.

4.3.1 Feature Extraction Agent

In the feature extraction agent, we use a pre-trained deep Convolutional Neural Network (CNN) to extract the image features which is a vector of the scores of each feature. In this work, we use Inception v3 developed by Google [1] which is the champion of 2015 ImageNet image recognition competition. It achieves 21.2% top-1 and 5.6% top-5 error for single frame evaluation on 1,100 general class of images. The number of network parameters may vary due to drop-out process and the upper limit is 25 million. Fig. 2.14 is a visualization of the model architecture. Section 1.7 has the detailed description about how the network works in extracting the features.

The output number of features is 2,048 for the Inception v3 which we believed to be adequate in capturing the feature information (including low, median and high level). In the dataset, the typical dimension for the images are $256 \times 256 \times n$ where n is the number of slices which ranges from 10 to 110 and most n is locate at 30 to 40. Thus, after feature

extraction, the 3-D images has largely reduced to 2-D feature maps with the dimension of $n \times 2048$ and ready to proceed to next agent: Image Selection.

4.3.2 Image Selection Agent

For image selection agent, each 2048-long feature vector will be evaluated by a universal approximator [55] to get a score of chance containing tumors or suspicious tissues. The slices with the top k scores will be sent to the final pathology diagnosis network.

In practical use we found that instead of picking single slices from the feature map, it is more efficient to select a window with length of k and send the reduced feature map $k \times 2048$ to the pathology prediction network. That is mainly because the tumors are 3-D objects which cross multiple slices. The window-based approach improves the performance of pathology prediction agent (about 10% increment of accuracy). In this case we choose $k = 3$ as an optimized length of window which balanced the performance and computational time.

Since the slices we picked are inside a window, the appearance of the image will be related as a sequence and so to the feature vectors. Thus it is possible to reduce the variables of the problem using the property. Bi-directional Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) cell [64] is an appropriate approach for the problem. The structure of RNN is shown in Fig. 4.3a. The output of both directions are aggregated to the final universal approximator to obtain the score. In this case the number of output nodes is 50 for each direction. Therefore the final universal approximator has 100 nodes and the final output is a score of risk which is evaluated by the slices images in the window.

By sliding the windows from the beginning, we obtain the $n-k$ scores with the convolution process. The score are identical if we slide from the other end because the symmetry of network structure. At the end of image selection agent, window with highest score will be send to the final pathology prediction agent.

4.3.3 Pathology Prediction Agent

Because the structure of the input is the window of feature map with size $k \times 2048$. Thus we can use the similar structure of RNN with LSTM cell as the image selection agent. But unlike image selection agent, the final layer of the prediction agent is a fully connected classifier which classify the result of the tumor: benign or malignant. Fig. 4.3b shows the structure of the prediction network.

4.4 Training and inferring

For all the three agents, except for the pre-trained CNN in feature extraction agent, the last two agents will be trained on our specific dataset.

We separate the total 76,795 samples into training set and test set where 90% (69,116) of the scans are training set and 10% of them are test set. We also keep the fraction of malignant results (12%) the same inside two datasets. The size of training set is adequate for the problem to avoid over-fitting. However, as we stated at the beginning, the information is sparse along the dataset. Carefully design of the training process is needed to perform a proper training.

The workflow of the training process is shown in Fig. 4.6, the training for image selection agent and pathology prediction agent are separate. Between each meta-step, the training of the two networks are independent. The basic workflow is to evaluate the performance of selection agent by the feedback given by the training of prediction agent. The score of each window is updated based on the prediction performance for each meta-step. By repeating the training over meta-steps, the score of each window inside are accumulated and the slices and will increase the performance over the meta-steps.

4.4.1 Bootstrapping Training

Because the samples are limited and it is a typical unbalanced training problem. Also, the variance caused by different appearance of breast are unknown. In order to simulated the real time scans according to the real distribution of MRI appearances. We choose bootstrapping sampling when we perform the training process.

Bootstrapping [109] is the name for any sampling with replacement. When we are feeding the batch (100 samples) inside each meta step, we sample evenly with replacement from both malignant and benign scans (50-50 each). This will ensure for each training step, the network receive the equal information from both sides and infer with no biases of prior malignant probabilities.

4.4.2 Score based Reinforcement Training

The lack of location information of tumors are the major problem for our training process. To overcome the problem and let the computer automatically detected the slices with tumor images. We introduce a reinforced learning process in automatically optimizing the slice window selection.

As shown in Fig. 4.6, the reinforcement learning is designed by the following algorithm:

1. Initially, for all slices windows j in scan i, the score $S_i(j) = 0$.

2. Pick the slice window $m \sim m + k$ with maximum L-2 norm of feature vector $F_i(j)$: $m_i = \underset{j}{\text{argmax}} \sum_{j..j+k} ||F_i(j)||_2$ as the initial window selection. Initial the score with $S_i(m_i) = 1$

3. Reset the weights for both image selection agent and pathology prediction agent.

4. Train the selection agent based on the score. The loss function used for the universal approximator is the least square score: $E_{UA} = \sum(S_i - \hat{(}S_i))^2$

5. Use re-trained selection network to obtain the estimated score of $\hat{S}_i(j)$ of each slice window j for every scan i.

6. Pick the slice window with the maximum estimated score $\hat{m}_i = \underset{j}{\text{argmax}} \hat{S}_i(j)$

7. Pass the selected slice window to prediction agent and train the prediction agent with the true class C_i of scan i. The loss function used for the training of prediction network is the softmax cross entropy function: $\xi(\hat{\mathbf{C}}, \mathbf{C}) = -\hat{C}log(C) - (1-\hat{C})log(1-C)$ where $C, \hat{C} = 1$ if the true/predicted scan result is malignant. Otherwise $C, \hat{C} = 0$

8. Update the score of each slice window x based on the selection result with Gaussian Kernel:

$$S_i(x) = (1-d)S_i(x) + R_i \exp[\frac{(x-\hat{m}_i)^2}{2\sigma^2}] \tag{4.1}$$

   Where $d = 0.1$ is a damping factor for previous memory. $\sigma = 3$ is a factor to control how wide the score distributed along the center. R_i is the prediction feedback given by the pathology prediction agent. If the pathology prediction agent gives a CORRECT prediction on malignant case: $R_i = 1$. Else if the pathology prediction agent gives a INCORRECT prediction on a benign case: $R_i = 0.5$. Otherwise, $R_i = 0$ for all other prediction.

9. Repeat steps 3 to 8 as a meta-step. Continuous training to accumulate the score of each slice window.

Ideally, by the process of learning, the selection agents will be accurate in selecting the slices window with potential risky tissues. However, the converge process is very slow without any prior information of what tumors looks like. Further discussion will be provided in the result section.

4.4.3 Inferring

Similar to the training process, the inference process follows the workflows below (Fig. 4.6b):

1. Use trained image selection agent to obtain the estimated score of $\hat{S}_i(j)$ of each slice window j for every scan i.
2. Pick the slice window with the maximum estimated score $\hat{m}_i = \underset{j}{\text{argmax}} \hat{S}_i(j)$
3. Pass the selected slice window to prediction agent and make the prediction of pathology result $\hat{C}_i$ of scan i.

Unlike the training process, we don't use the softmax function for inferring the final result. Instead, we use a controlled threshold to make our inference more flexible for different scenarios.

The output of inference network are two digits (l_0, l_1) which represent the log-likelihood of two classes (benign and malignant correspondingly). We set up a thresholds ζ and mark all the scans with $l_1 - l_2 > \zeta$ as malignant. By controlling the threshold ζ, we are able to obtain a curve called Receiver operating characteristic (ROC) [110] to proceed to further evaluation of the model.

In order to increase the sensitivity of the result and avoid any missing malignant tissue, practically the result will be obtained by comparing all the slice windows with top 3 scores. If all the prediction result gives negative (benign), then the scan is labeled as benign.

4.5 Results

We test our result on both T1 and T2 sequences. We set up three tasks to test our performance of the model:

1. Based on all scans, predicting the pathology result before the doctor given the BIRADS of each scan.

2. Predicting the pathology result after the doctor given the BIRADS of each scan. Namely, we train the networks based on the scans with $BIRADS \geq 4$ and try to assist the doctor for the further biopsy result.

3. Predicting the BIRADS class rated by the pathologists. Instead of the pathology result, we set up the two classes by $BIRADS \geq 4$ and $BIRADS < 4$. This task is to evaluate how close the networks can simulate the rating of pathologists.

To evaluate the result, we use Receiver operating characteristic (ROC) curve. The ROC curve is a convex curve plotted on True Positive Rate (TPR) and False Positive Rate (FPR) (Please see the table 4.1 for the detailed definition of TPR and FPR). To evaluate the performance of the model, we use the AUROC (Area under the ROC curve) as a proxy of curve convexity. The larger ROC means the model reduces the false positive rate while keeping a high sensitivity (TPR) in detection.

The performance of pathologist also can be marked as a point in the TPR-FPR space. According to the labeling process of BIRADS and criteria of doing biopsy. We can assume that the prediction of biopsy result made by pathologists is by BIRADS. If $BIRADS \geq 4$, we assume the doctors made the prediction of 'malignant' on the scan. Because malignant tumors can only be detected after they send to biopsy, the doctors always get the TPR = 1. On the other hand, among all the 9,078 scans with $BIRADS \geq 4$, there are 4,625 (50.9%) of them are malignant. Thus the FPR rate for the doctors are 0.509.

Because the first-type error (miss any malignant case) is fatal in clinic, our goal for the network models is to assist the doctors to filter the scans to reduce the FPR rate to save the cases send to biopsy while keeping the TPR equals to 1. Such criteria will result a very restrict threshold ζ in making the prediction.

4.5.1 Feature Properties

After the feature extraction agent, all images are compressed to vectors with 2048 features. Generally, the variables are not independent due to the limited degree of freedom in the MRI scans with same type. Fig. 4.4 plot the empirical Pearson Correlation Matrix between features (f_i, f_j) and clustered the nodes with their intensity of correlation / anti-correlation. Four major highly correlated clusters have emerged. Although the meaning and the exact nature of these features are difficult to interpret, the correlation implies the feature map can be reduced to a more compact model with less parameters.

4.5.2 Training Result

During the training, in order to correctly evaluate the performance over meta-steps. We choose cross validation [111] methods to monitor the convergence of the training function. In each meta-step, we randomly hold out two batches (200 samples with 100 in each class) as the validation data. After each cycle of meta-step, we evaluate the validation accuracy (see table 4.1 for definition) and loss function of pathology prediction agent. The result of the three tasks are shown in Fig 4.5.

The convergence of loss function and accuracy is very slow and random due to the randomness of the selection process. We could observe several step-like increments during the training which implies the progress made by reinforced learning processes. However, due to the limitation of our computation time and resources, we can not obtain more results after the future steps.

4.5.3 Testing Result

We plot the ROC curves for the three tasks based on T2 sequences. The results for top-1 slice inference are shown in Fig. 4.7. The corresponding area under ROC curve (AUROC)

and the test accuracy is presented in Table 4.2. The result of a single slice window inference (Fig. 4.7a to c) show the network performed best for task 1 with an overall accuracy of 65%. However, the accuracy does not meet the requirement of a reliable predictor. Accuracy will increase if we aggregate the prediction result on the window slice with top 3 scores (the output prediction is taken from the window slice with largest differences in log-likelihood: $(l_1 - l_0)$). The accuracy will be slightly increase due to the increment of sensitivity. However, as we add the number of candidate slice windows to 5, the model become oversensitive which results in a drop of accuracy. Moreover, from the right end point of ROC curve, TPR drops immediately when FPR drops. These phenomena implies the pathology network is not correctly trained due to a mistaken selection of slices which is very likely exist.

The poor performance of task 2 implies the relations between pathology result and MRI scan visual information can not be easily identified. If we plot the ROC over a subset of test samples with $BIRAD = n$ (Fig. 4.7d), we can find that the algorithm is more sensitive to the samples with high BIRADS value. This is intuitive because a higher BIRADS value means there is more evidence for pathologists to believe the tumors are malignant. It implies the prediction network actually capture such evidence. Nevertheless, these evidence are not very outstanding over the noise or image feature variance.

Task 3 is a reference to compare the differences between the machine and doctors in rating the risk of scans. We notice that the model performs better in Task 1 (direct approach) than Task 3 (mimic doctors). This may indicate the intrinsic differences in capturing the evidences of pathology results.

We also compare the result of Task1 between different MRI sequences. Fig. 4.8 is the comparison between T1 images with contrast and T2. T1 contrast are slightly out-performed to T2 with an slightly increment of ROC of 5%. It suggest the highlights of tumor location benefits the prediction result.

4.6 Discussion and conclusion

The preliminary analysis above demonstrates the ability of the network in discovering the signs for different types of tumor. However, such ability is not reliable because the prediction network is not well trained. The critical reason for the ill trained network is that we have no prior information of the tumor locations. In the areas outside tumors, malignant scans looks the same with benign scans. Misfed slice windows will result a poor variable space for the training set. Although Reinforcement Learning can be helpful in filtering the tumor images, the efficiency of the model is not guaranteed.

Several approaches can be applied to improve the performance of the model.

1. Brutally extend the computational ability to accelerate the training step in order to get more meta-steps of reinforcement learning.

2. Boost the learning accuracy by introducing the prior of tumor images. MSKCC has been working on labeling the rough location of tumor for 1000 scans. These scans with tumor will be a good bootstrap training set.

3. Make full use of T1 post contrast sequences. Although the information of the water signal will be lost in T1 sequences, T1 are mode widely used among pathologists in predicting the pathology result. However, T1 post contract images are generated in sequence after screening, the relations between images in different time step also matters in identifying the area and type of tumor. Therefore the complexity of the problem in capturing the image signals will become more complicated which requires an other carefully designed models.

The project is still on-going. It will continuously updated when new information is available.

Table 4.1: Confusion Matrix used for

		Biopsy Results: Malignant	Biopsy Results: Benign
Predicted Results	Malignant	True Positive (TP)	False Positive (FP)
	Benign	False Negative (FN)	True Negative (TN)

Table 4.2: Test result summary for three tasks

	Top-1 AUROC	Top-1 Accuracy	Top-3 Accuracy	Top-5 Accuracy
Task 1	0.688	0.630	0.660	0.575
Task 2	0.574	0.550	0.585	0.545
Task 3	0.647	0.604	0.632	0.630

True Positive Rate $TPR = \frac{TP}{TP+FN}$

False Positive Rate $FPR = \frac{FP}{TP+FN}$

Accuracy $AC = \frac{TP+TN}{TP+TN+FP+FN}$

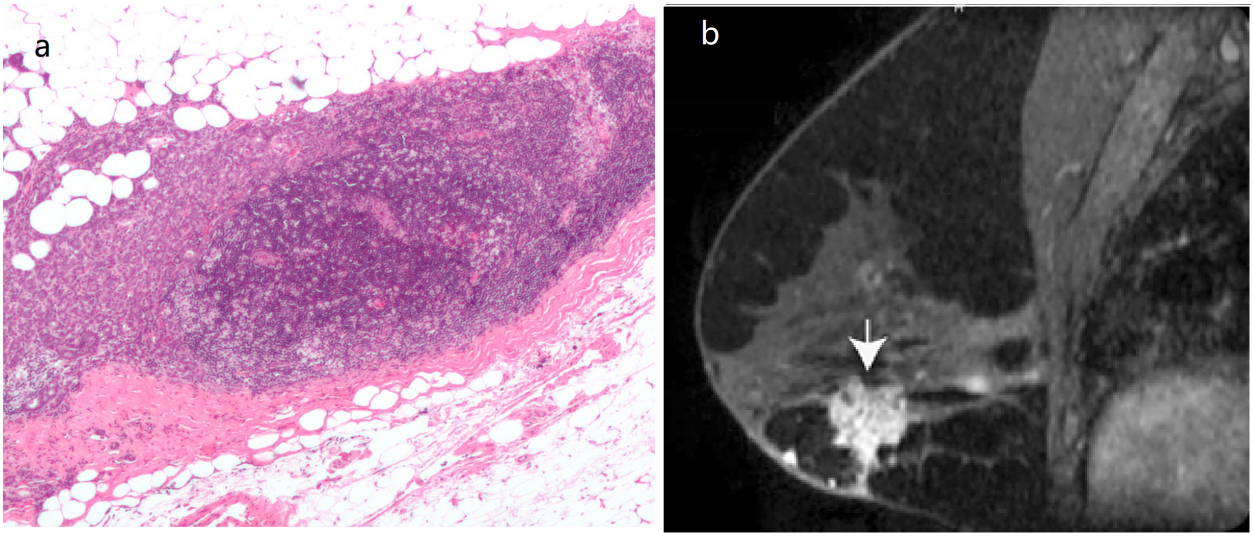

Figure 4.1: Example of micrograph and MRI image for a breast tumor

The example shows inferring process for a slice window size of 3. Feature vectors of slices are sequently input to an RNN with LSTM cell. The RNN is bi-directed, the final output of both direction are h_- and h_+ (50 nodes for each). Hidden nodes h_- and h_+ are aggregated and input to the next level: a) Universal Approximator for Image selection (scoring) network **b)** Pathology Prediction Network.

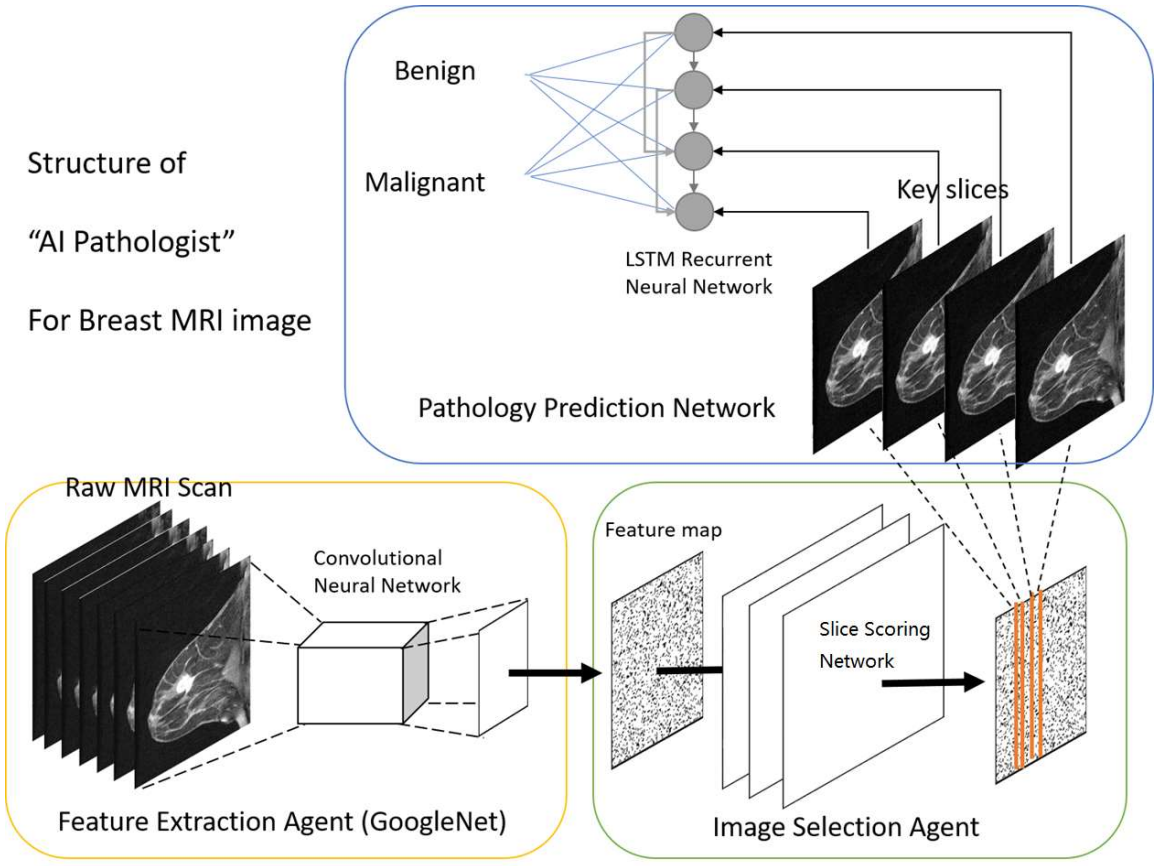

Figure 4.2: Workflow of deep learning in predicting the pathology result from MRI scan

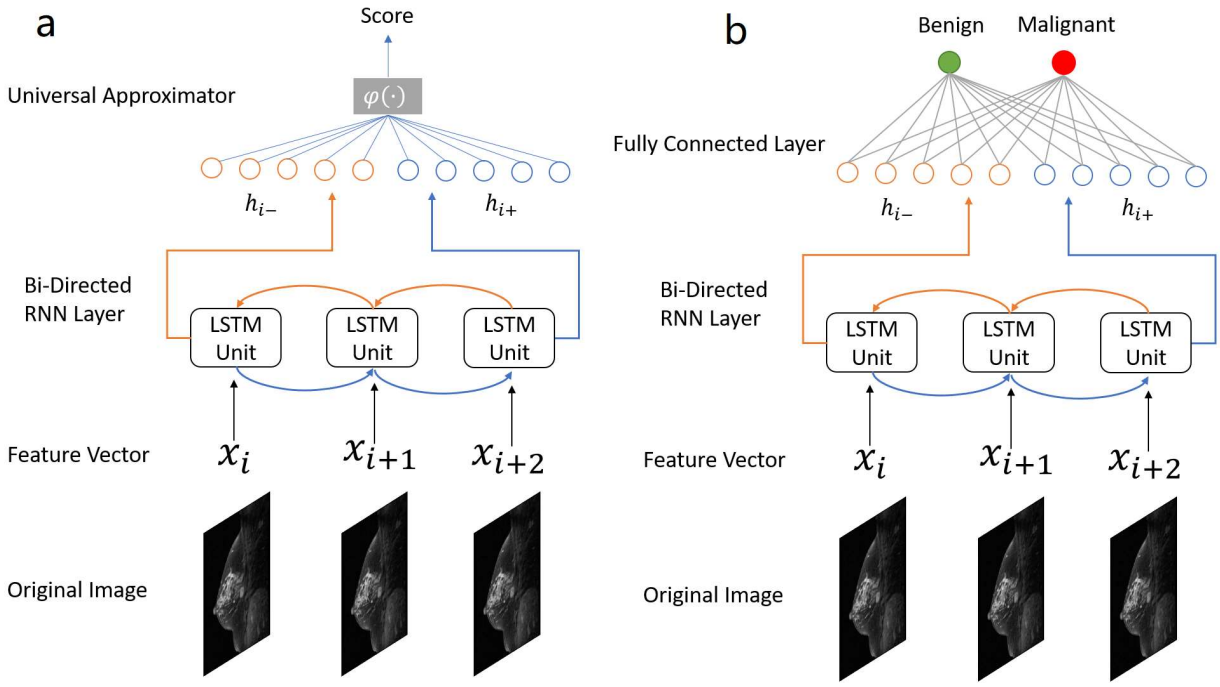

Figure 4.3: Structure of Image Selection Network and Pathology Prediction Network

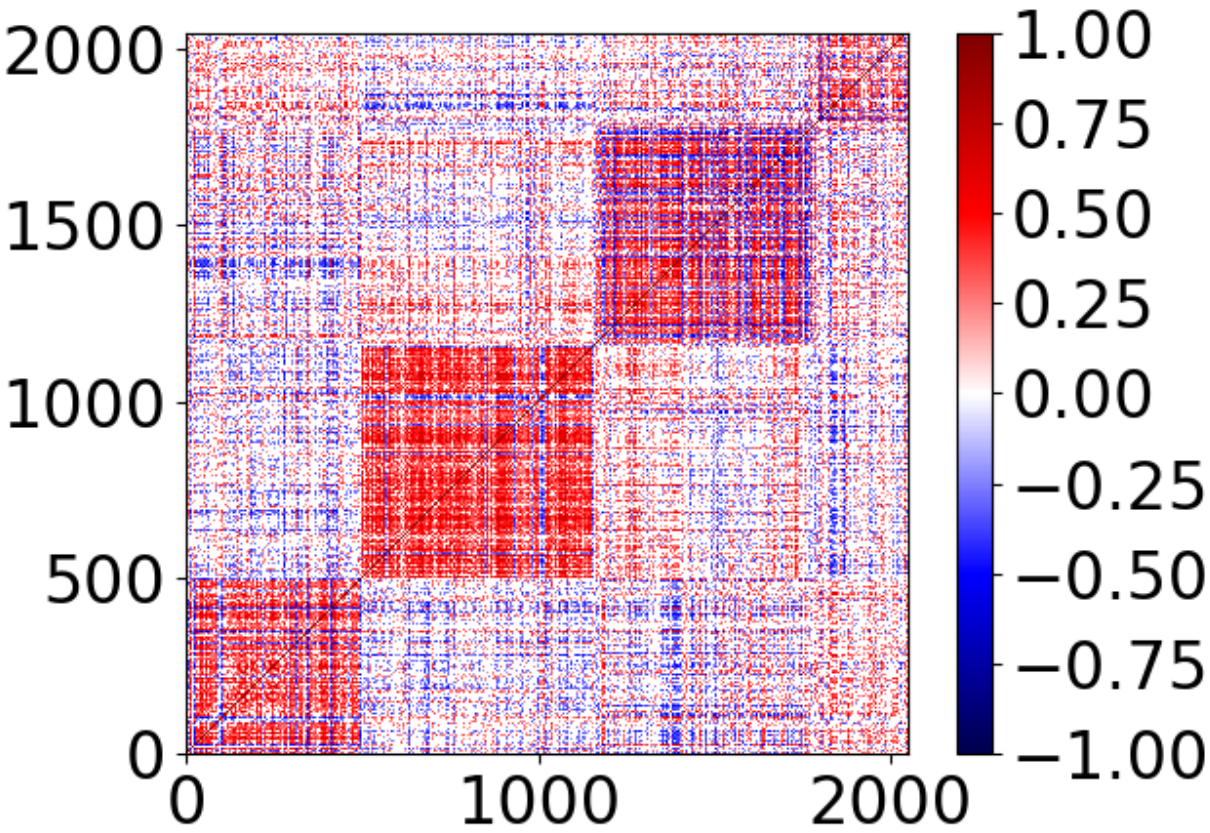

Figure 4.4: Structure of Image Selection Network and Pathology Prediction Network

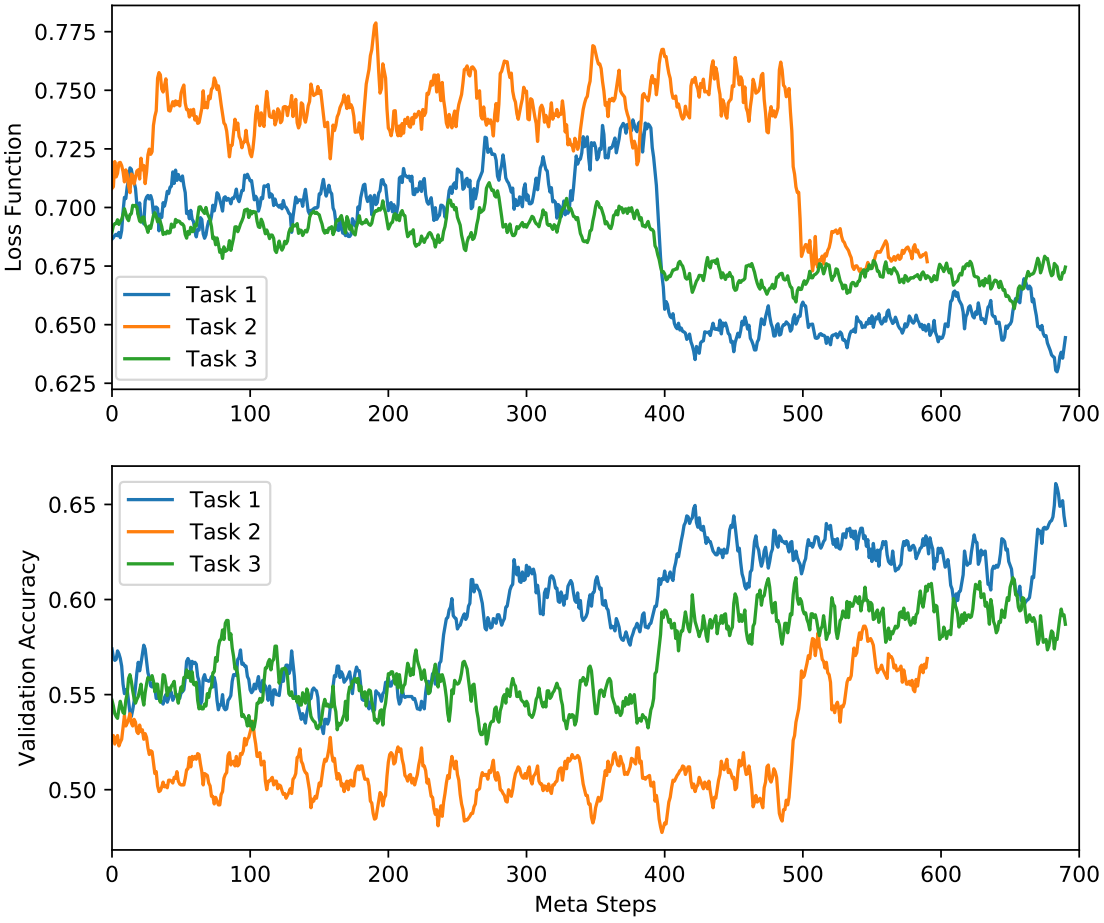

Figure 4.5: Loss function and validation accuracy change with training steps For task 1 and task 3, we trained for 700 meta-steps while for task 2 is 600.

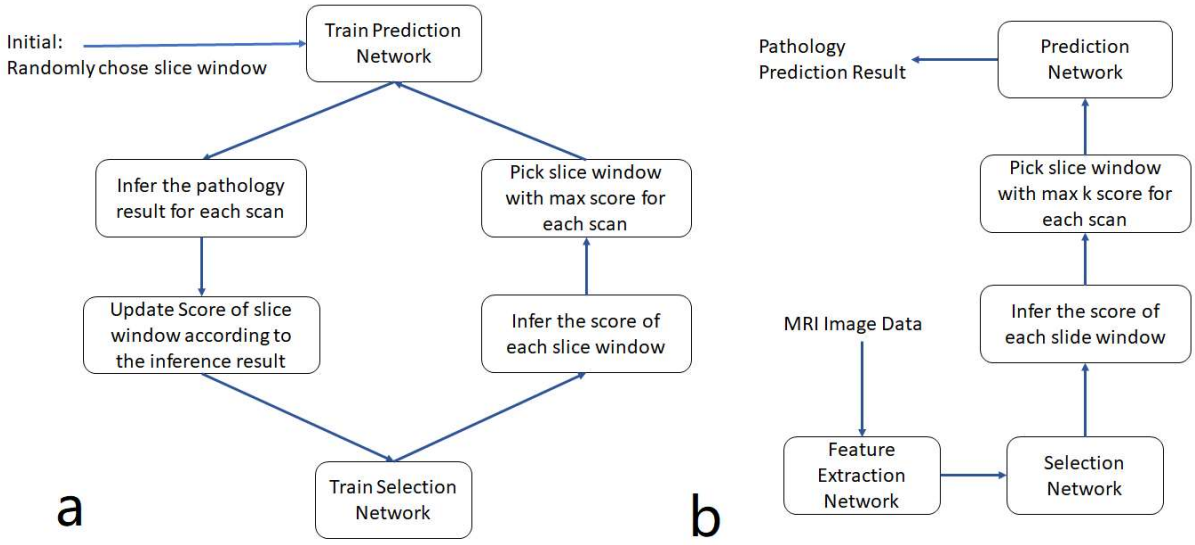

Figure 4.6: Training and Inferring workflow

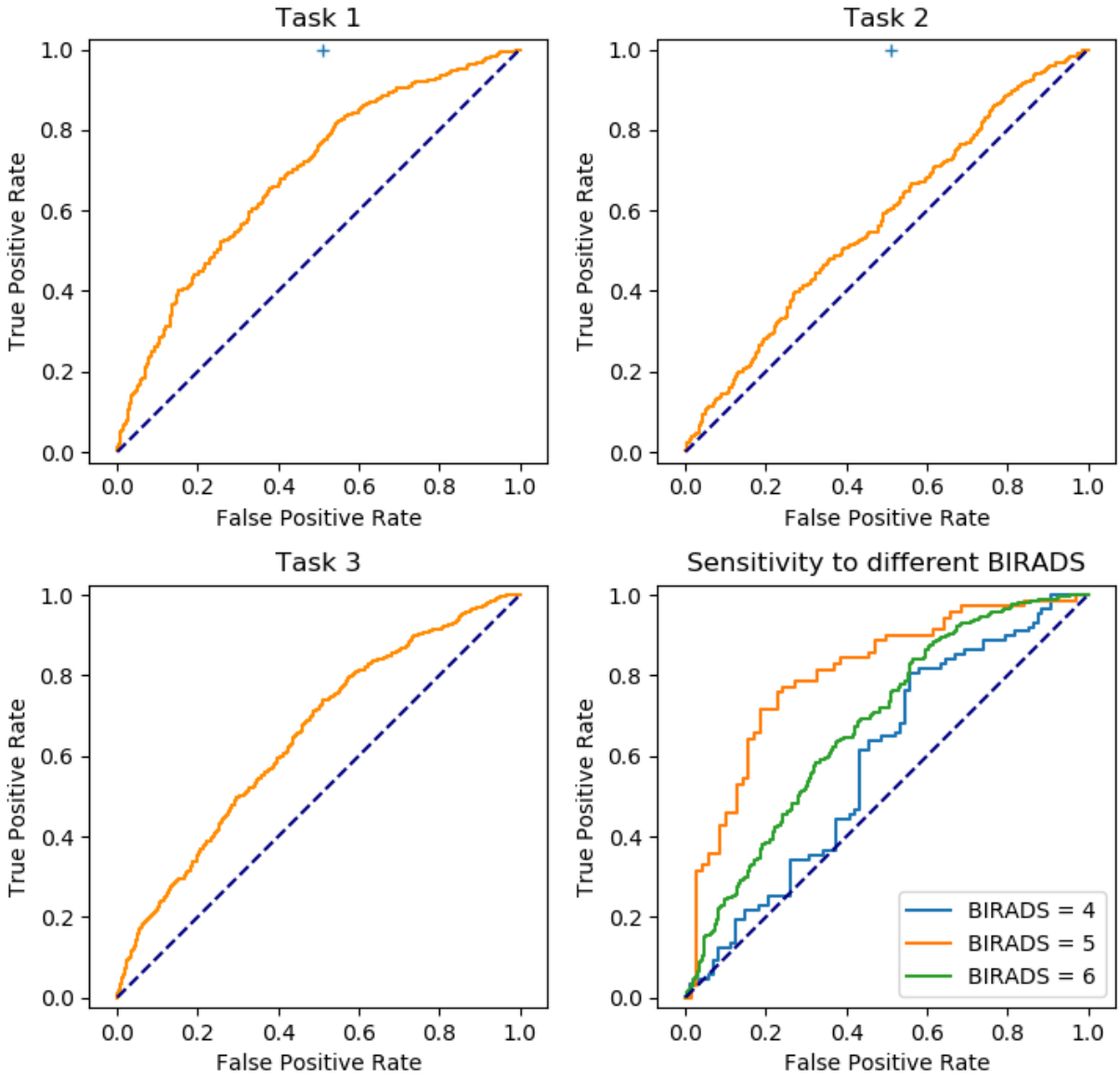

Figure 4.7: ROC curves for different tasks

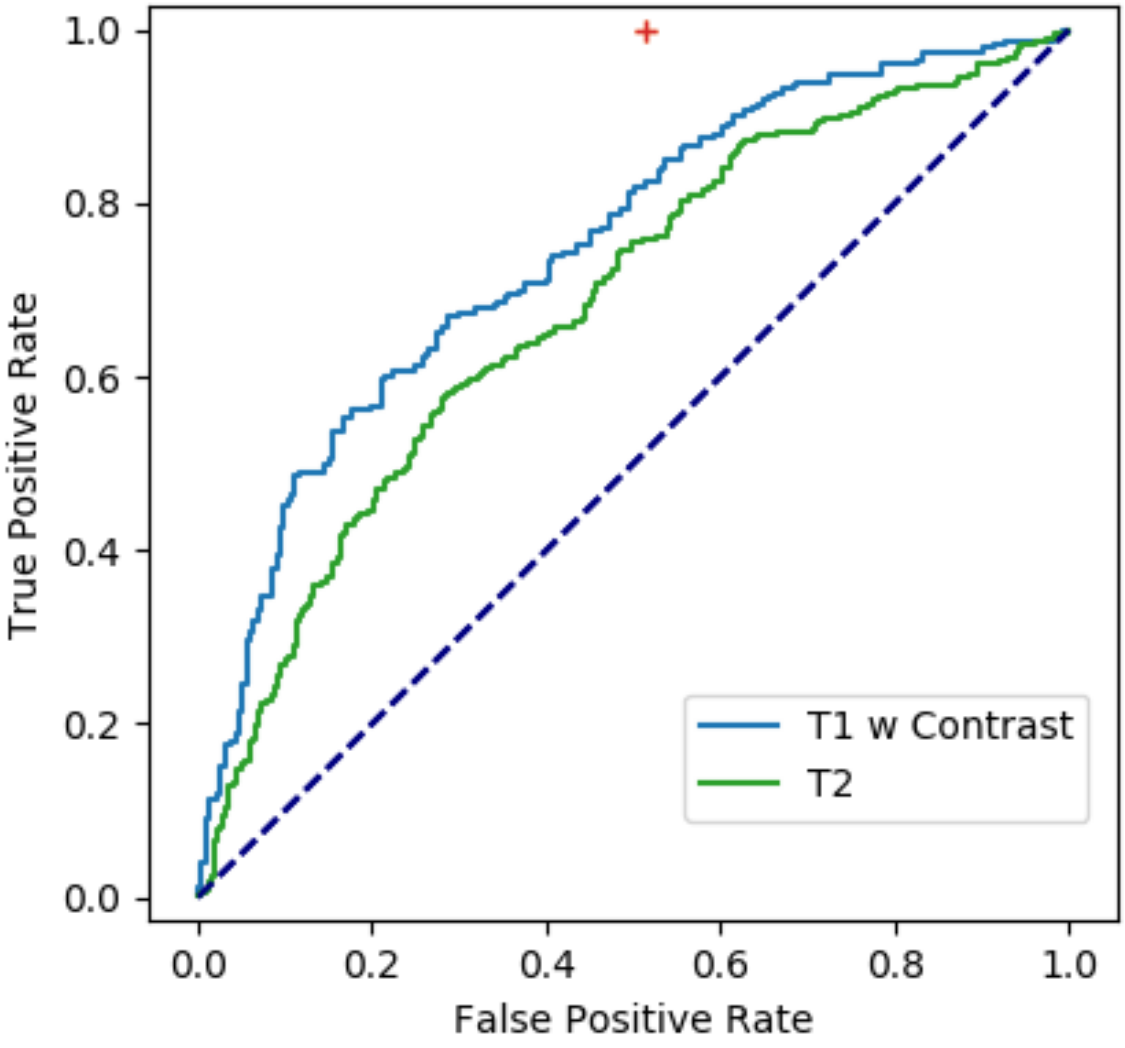

Figure 4.8: ROC curves of T1 with contrast and T2

Chapter 5

Conclusion

From all the analysis above, we can conclude network-based statistical inference models are powerful tools for solving a varies problem in the real world. It is advantage in representing the variable correlation make it especially suitable for high dimensional but sparse problems like image recognition and big data mining. However, network based models are facing bottlenecks in nearing future due to several reasons. Some of the major reasons are:

1. As the deep learning models become deeper and wider. Understanding and reasoning of the model become extremely difficult. Just like the features of CNN, for most of the nodes inside deep learning models has no corresponding meanings related to logical recognition. Such complexity result in the optimization and debugging of the network is complicated and resources consuming work, both for time and computational.

2. The learning efficiency for network based model is extremely low compare with human. For example, CNN requires millions of images to make it manage to recognize objects. However, it only require few images for human. The reason why CNN has such poor efficient is that CNN is missing some logical relations between different part of objects. For example, Both pictures in Fig. 5.1 are categorized as human face for CNN because

they just record the feature rather than record their spatial relations. Also, to infer the view from another angle of an 3D object is also difficult for CNN's where human, on the other hand, can easily done it.

3. Although the network based network inference model can be applied in many occasions. However, for the problem with unstructured data where the input variable size may varies and some variables may be missing. The fix-structured network models will not be suitable to solve the problem. Namely, networks are lack of ability to "reasoning".

Some major breakthroughs has been made to overcome the limitation of network inference model. One of the promising solution is the use of capsule by Hinton [112]. In this work Hinton mimic the activity of human's cortex in brain deeply. The work pack a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity such as an object or an object part into a capsule. Active capsules at one level make predictions, via transformation matrices, for the instantiation parameters of higher-level capsules. When multiple predictions agree, a higher level capsule becomes active. By implementing such complex interaction, capsules are able to perform better then CNN in recognizing the overlap digits which shows the sign of reasoning.

Although the research on capsule is just started, the development of more powerful network based tools never stops. We are expecting a brighter future on network applications with unprecedented possibilities.

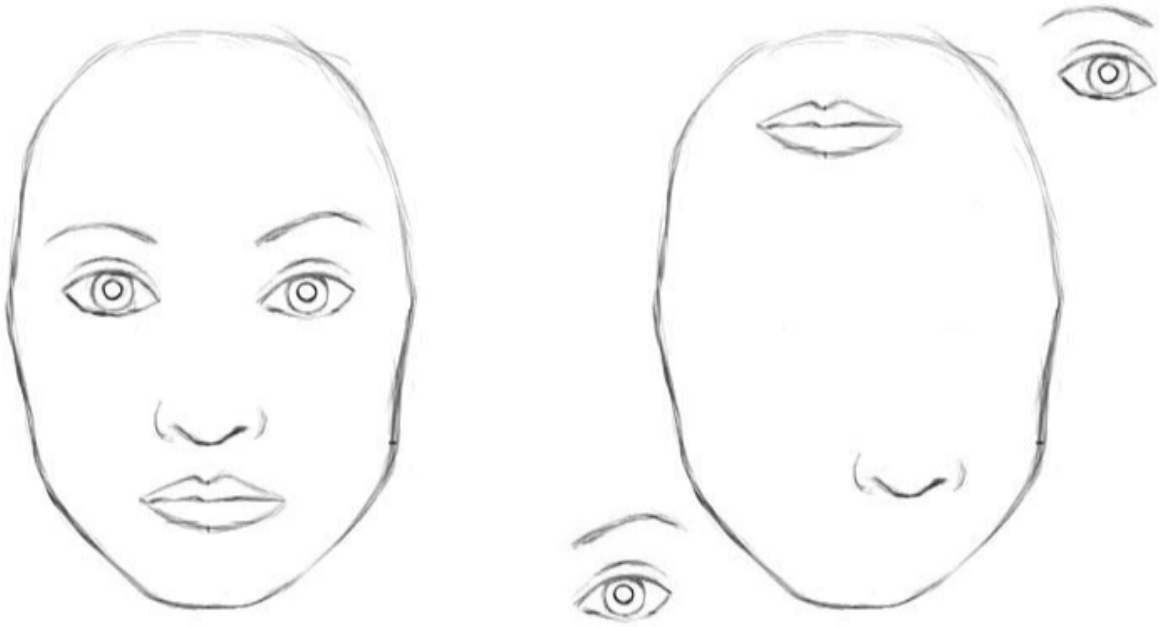

Figure 5.1: An example showing how CNN fails

Chapter 6

Appendix: Code and Files

Code and related files for Chapter 4 are available in the following link:

https://github.com/shaojunluo/Disseration

The description of each files are listed below. Due to privacy reason, data files are not listed in the repository:

6.1 Preprocessing

1. **read_dicom.py** Read all .dicom files and extract information from DICOM files (date-time, type, subject ID, dimension etc.). Output as a table with list of folders and corresponding information.

2. **read_birads.py** Clean the BIRADS data from folder table generated by read_dicom.py. Filter the data with valid image and text information.

3. **read_biopsy.py** Combine the biopsy result table with DICOM file according to the subject ID and date of completion. Generate and validate the final labeled table for each scan.

4. **filter_list.py** Filter the specific sequence of MRI scans (T1, T2, etc) according to the key words written in the DICOM file.

6.2 FeatureExtraction

extract_features.py Feature Extraction Agent. Extract features of DICOM files from pre-trained Inception v3 CNN [1]. It automatically download and run the models on MRI images stored as DICOM files in a folder (1 scan per folder). The output of each scan is the feature map matrix n*2048 where n is the number of DICOM files and 2048 is the number of feature extracted.

6.3 Training-and-Testing

1. **funclib.py** Customized function library for training and testing.

2. **rnn_reinforce_v3_training.py** Train the Selection Agent and Prediction Agent using the reinforcement learning describe in Chap 4. The training is deployed on the CPU machine with tensorflow. Change of loss function and training accuracy are stored when training.

3. **Inference.py** Inference the result of trained Selection Agent and Prediction Agent. ROC curve and confusion matrix are generated during the inference to evaluate the performance of trained model.

Bibliography

[1] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[2] Seref Sagiroglu and Duygu Sinanc. Big data: A review. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 42–47. IEEE, 2013.

[3] Vasant Dhar. Data science and prediction. *Communications of the ACM*, 56(12):64–73, 2013.

[4] Christopher J Shallue and Andrew Vanderburg. Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *The Astronomical Journal*, 155(2):94, 2018.

[5] Sankalp Gilda, Jian Ge, et al. Parameterization of marvels spectra using deep learning. In *American Astronomical Society Meeting Abstracts*, volume 231, 2018.

[6] Scott J Lusher, Ross McGuire, René C van Schaik, C David Nicholson, and Jacob de Vlieg. Data-driven medicinal chemistry in the era of big data. *Drug discovery today*, 19(7):859–868, 2014.

[7] Hayit Greenspan, Bram van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[8] Matthew A Waller and Stanley E Fawcett. Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management. *Journal of Business Logistics*, 34(2):77–84, 2013.

[9] W Lawrence Neuman. *Social research methods: Qualitative and quantitative approaches*. Pearson education, 2013.

[10] David Lazer, Alex Sandy Pentland, Lada Adamic, Sinan Aral, Albert Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

[11] Heikki Mannila. Data mining: machine learning, statistics, and databases. In *Scientific and Statistical Database Systems, 1996. Proceedings., Eighth International Conference on*, pages 2–9. IEEE, 1996.

[12] Chen Chi Hau. *Handbook of pattern recognition and computer vision*. World Scientific, 2015.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436, 2015.

[14] Calyampudi Radhakrishna Rao, Calyampudi Radhakrishna Rao, Mathematischer Statistiker, Calyampudi Radhakrishna Rao, and Calyampudi Radhakrishna Rao. *Linear statistical inference and its applications*, volume 2. Wiley New York, 1973.

[15] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[16] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

[17] Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[18] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 7(8), 2015.

[19] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016.

[20] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[21] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[22] Filippo Radicchi and Claudio Castellano. Beyond the locally treelike approximation for percolation on real networks. *Physical Review E*, 93(3):030302, 2016.

[23] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[24] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[25] Kenneth J Arrow and Leonid Hurwicz. Reduction of constrained maxima to saddle-point problems. In *Traces and Emergence of Nonlinear Programming*, pages 61–80. Springer, 2014.

[26] Evgenii Solomonovich Levitin, Aleksei Alekseevich Milyutin, and Nikolai Pavlovich Osmolovskii. Conditions of high order for a local minimum in problems with constraints. *Russian Mathematical Surveys*, 33(6):97, 1978.

[27] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[28] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

[29] Christian Robert. Machine learning, a probabilistic perspective, 2014.

[30] Marc Mézard and Giorgio Parisi. The bethe lattice spin glass revisited. *The European Physical Journal B-Condensed Matter and Complex Systems*, 20(2):217–233, 2001.

[31] Simon Haykin and Neural Network. A comprehensive foundation. *Neural networks*, 2 (2004):41, 2004.

[32] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[33] Leann Myers and Maria J Sirois. Spearman correlation coefficients, differences between. *Wiley StatsRef: Statistics Reference Online*, 2006.

[34] Dietrich Stauffer and Ammon Aharony. *Introduction to percolation theory*. CRC press, 1994.

[35] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[36] Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In *Advances in neural information processing systems*, pages 3165–3173, 2013.

[37] Paul S Bradley, Olvi L Mangasarian, and W Nick Street. Clustering via concave minimization. In *Advances in neural information processing systems*, pages 368–374, 1997.

[38] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.*, 2008(10): P10008, 2008.

[39] Mark EJ Newman. Analysis of weighted networks. *Phys. Rev. E*, 70(5):056131, 2004.

[40] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders in complex networks. *Nature. Phys.*, 6(11):888–893, 2010.

[41] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001.

[42] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 422, Stanford InfoLab, 1998.

[43] Flaviano Morone and Hernán A Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563):65–68, 2015.

[44] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 137–146. ACM, 2003.

[45] Nathan Eagle, Michael Macy, and Rob Claxton. Network diversity and economic development. *Science*, 328(5981):1029–1031, 2010.

[46] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[47] Tsung-Dao Lee and Chen-Ning Yang. Statistical theory of equations of state and phase transitions. ii. lattice gas and ising model. *Physical Review*, 87(3):410, 1952.

[48] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Generalized belief propagation. In *Advances in neural information processing systems*, pages 689–695, 2001.

[49] Emile Aarts and Jan Korst. Simulated annealing and boltzmann machines. 1988.

[50] Toshiyuki Tanaka. Mean-field theory of boltzmann machine learning. *Physical Review E*, 58(2):2302, 1998.

[51] Hsin Chen and Alan F Murray. Continuous restricted boltzmann machine with an implementable training algorithm. *IEE Proceedings-Vision, Image and Signal Processing*, 150(3):153–158, 2003.

[52] Loyce Adams. M-step preconditioned conjugate gradient methods. *SIAM Journal on Scientific and Statistical Computing*, 6(2):452–463, 1985.

[53] Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8-9):686–713, 2006.

[54] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[55] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[56] Bart Kosko. Fuzzy systems as universal approximators. *IEEE transactions on computers*, 43(11):1329–1333, 1994.

[57] Rob A Dunne and Norm A Campbell. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, volume 181, page 185, 1997.

[58] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[59] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[60] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[61] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[62] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[63] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[64] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[65] Roger A Horn. The hadamard product. In *Proc. Symp. Appl. Math*, volume 40, pages 87–169, 1990.

[66] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45 (2):167–256, 2003.

[67] Alessandro Vespignani and Guido Caldarelli. *Large Scale Structure and Dynamics of Complex Networks: From information technology to finance and natural science*. World scientific, 2007.

[68] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge University Press, Cambridge, UK, 1994.

[69] Mark S Granovetter. The strength of weak ties. *Am. J. Sociol.*, 78(6):1360–1380, 1973.

[70] Mark Granovetter. The impact of social structure on economic outcomes. *J. Eco. Perspect.*, 19(1):33–50, 2005.

[71] Vivek K Singh, Laura Freeman, Bruno Lepri, and Alex Sandy Pentland. Predicting spending behavior using socio-mobile features. In *2013 Int. Conf. on Social Computing (SocialCom)*, pages 174–179. IEEE, 2013.

[72] Walter W Powell and Laurel Smith-Doerr. Networks and economic life. *The handbook of economic sociology*, 368:380, 1994.

[73] David Strang and Sarah A Soule. Diffusion in organizations and social movements: From hybrid corn to poison pills. *Annu. Rev. of Sociol.*, 24(1):265–290, 1998.

[74] Ronald S Burt. *Structural holes: The social structure of competition*. Harvard university press, Cambridge, MA, 2009.

[75] Scott E Page. *The difference: How the power of diversity creates better groups, firms, schools, and societies*. Princeton University Press, Princeton, NJ, 2008.

[76] Roberto M Fernandez and Nancy Weinberg. Getting a job: networks and hiring in a retail bank. *Stanford GSB Research Paper Series*, 1382:1, 1996.

[77] Catherine Zimmer. Entrepreneurship through social networks. *The art and science of entrepreneurship*, pages 3–23, Ballinger, Cambridge, MA, 1986.

[78] Linton C Freeman. Centrality in social networks conceptual clarification. *Soc. Networks*, 1(3):215–239, 1978.

[79] Jameson L Toole, Yu-Ru Lin, Erich Muehlegger, Daniel Shoag, Marta C González, and David Lazer. Tracking employment shocks using mobile phone data. *J. R. Soc. Interface*, 12(107):2015.0185, 2015.

[80] Marc-David L Seidel, Jeffrey T Polzer, and Katherine J Stewart. Friends in high places: The effects of social networks on discrimination in salary negotiations. *Admin. Sci. Q.*, 45(1):1–24, 2000.

[81] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proc. of the 17th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 1082–1090. ACM, 2011.

[82] Santi Phithakkitnukoon, Zbigniew Smoreda, and Patrick Olivier. Socio-geography of human mobility: A study using longitudinal mobile phone data. *PloS One*, 7(6): e39253, 2012.

[83] Pierre Deville, Chaoming Song, Nathan Eagle, Vincent D Blondel, Albert-László Barabási, and Dashun Wang. Scaling identity connects human mobility and social interactions. *Proc. Natl. Acad. Sci.*, 113:7047–7052, 2016.

[84] Luca Pappalardo, Maarten Vanhoof, Lorenzo Gabrielli, Zbigniew Smoreda, Dino Pedreschi, and Fosca Giannotti. An analytical framework to nowcast well-being using mobile phone data. *Int. J. Data. Sci. Anal.*, 2(1-2):75–92, 2016.

[85] Wei Pan, Gourab Ghoshal, Coco Krumme, Manuel Cebrian, and Alex Pentland. Urban characteristics attributable to density-driven tie formation. *Nature Comm.*, 4:1961, 2013.

[86] Thoralf Gutierrez, Gautier Krings, and Vincent D Blondel. Evaluating socio-economic state of a country analyzing airtime credit and mobile phone datasets. *Preprint at https://arxiv.org/abs/1309.4496*, 2013.

[87] Albert Ali Salah, Bruno Lepri, Fabio Pianesi, and Alex Sandy Pentland. Human behavior understanding for inducing behavioral change: application perspectives. In *(Eds. Salah, A., Lepri, B.) Int. Workshop on Human Behavior Understanding*, pages 1–15. Springer-Verlag, 2011.

[88] Adeline Decuyper, Alex Rutherford, Amit Wadhwa, Jean-Martin Bauer, Gautier Krings, Thoralf Gutierrez, Vincent D Blondel, and Miguel A Luengo-Oroz. Estimating food consumption and poverty indices with mobile phone data. Technical report, United Nations Global Pulse, New York, 2014.

[89] J. Blumenstock. Calling for better measurement: Estimating an individuals wealth and well-being from mobile phone transaction records. In *Proc. 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. ACM, 2014.

[90] J-P Onnela, Jari Saramäki, Jorkki Hyvönen, György Szabó, David Lazer, Kimmo Kaski, János Kertész, and A-L Barabási. Structure and tie strengths in mobile communication networks. *Proc. Nat. Acad. of Sci.*, 104(18):7332–7336, 2007.

[91] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[92] Nathan Eagle, Alex Sandy Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proc. Nat. Acad. of Sci.*, 106(36):15274–15278, 2009.

[93] Joseph E Stiglitz. *The price of inequality: How today's divided society endangers our future*. W. W. Norton & Company, New York, NY, 2012.

[94] Albert R Wildt and Olli Ahtola. *Analysis of covariance*, volume 12. Sage Publications, Beverly Hills, CA, 1978.

[95] Sen Pei and Hernán A Makse. Spreading dynamics in complex networks. *J. Stat. Mech. Theor. Exp.*, 2013(12):P12002, 2013.

[96] BWKP Stewart, Christopher P Wild, et al. World cancer report 2014. *Health*, 2017.

[97] AB Miller, BFAU Hoogstraten, MFAU Staquet, and A Winkler. Reporting results of cancer treatment. *cancer*, 47(1):207–214, 1981.

[98] JENNIFER L Kelsey. A review of the epidemiology of human breast cancer. *Epidemiologic reviews*, 1:74–109, 1979.

[99] Marc J Van De Vijver, Yudong D He, Laura J Van't Veer, Hongyue Dai, Augustinus AM Hart, Dorien W Voskuil, George J Schreiber, Johannes L Peterse, Chris Roberts, Matthew J Marton, et al. A gene-expression signature as a predictor of survival in breast cancer. *New England Journal of Medicine*, 347(25):1999–2009, 2002.

[100] Deborah Ford, DF Easton, M Stratton, S Narod, D Goldgar, P Devilee, DT Bishop, B Weber, G Lenoir, J Chang-Claude, et al. Genetic heterogeneity and penetrance analysis of the brca1 and brca2 genes in breast cancer families. *The American Journal of Human Genetics*, 62(3):676–689, 1998.

[101] RW Noyes, AT Hertig, and J Rock. Dating the endometrial biopsy. *Obstetrical & Gynecological Survey*, 5(4):561–564, 1950.

[102] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[103] Wentao Zhu, Chaochun Liu, Wei Fan, and Xiaohui Xie. Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification. *arXiv preprint arXiv:1801.09555*, 2018.

[104] Guotai Wang, Wenqi Li, Sebastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. *arXiv preprint arXiv:1709.00382*, 2017.

[105] Evangelia I Zacharaki, Sumei Wang, Sanjeev Chawla, Dong Soo Yoo, Ronald Wolf, Elias R Melhem, and Christos Davatzikos. Classification of brain tumor type and grade using mri texture and shape in a machine learning scheme. *Magnetic resonance in medicine*, 62(6):1609–1618, 2009.

[106] Mehmet Ufuk Dalmış, Geert Litjens, Katharina Holland, Arnaud Setio, Ritse Mann, Nico Karssemeijer, and Albert Gubern-Mérida. Using deep learning to segment breast and fibroglandular tissue in mri volumes. *Medical physics*, 44(2):533–546, 2017.

[107] Emilio Garcia, Renato Hermoza, Cesar Beltran Castanon, Luis Cano, Miluska Castillo, and Carlos Castanñeda. Automatic lymphocyte detection on gastric cancer ihc images using deep learning. In *Computer-Based Medical Systems (CBMS), 2017 IEEE 30th International Symposium on*, pages 200–204. IEEE, 2017.

[108] RM Henkelman, GJ Stanisz, and SJ Graham. Magnetization transfer in mri: a review. *NMR in Biomedicine*, 14(2):57–64, 2001.

[109] Bradley Efron and Robert Tibshirani. Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

[110] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[111] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[112] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017.