Random Walk Algorithms and Application: A Survey

A
School of Electrical and
Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: http://www.michaelshell.org/contact.html

B
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

C
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678-2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—Random walk has been widely used in the computer science community. Many researchers have exploited random walk approach achieving great result in almost all the areas. This paper presents a survey of random walk including quantum walk. We introduce important classical proximity measures and algorithms based on random walk. We also present a quantum view of random walk. We also illustrate different behaviour and properties between random walk and quantum walk. Finally, we discuss the obstacles in front of us when we exploit random approaches.**

1. Introduction

Random walk has a very long history. It was first introduced by Pearson in 1905 [1]. Since it was presented, mathematicians, physicians and computer scientists have do much research on it. In the field of mathematics, Spitzer [2] give a complete review of random walk for mathematic researchers and clearly state the mathematic principles of random walk. In the field of physics, there is adequate literature summarizing the random walk and quantum walk elements in physics [3], [4], [5], [6]. Random walk and quantum walk have been used broadly in computer science community during the past few years. Many researchers have exploited random walk approach in almost all kinds of areas in computer science. No matter in computer vision, recommender system or semi-supervised learning, we can all find that random walk approach gives us a good perspective to solve practical problems. In the field of complex social network analysis, Sarkar [7] gives us a survey of random walks' application and tests some random walk approaches used in social network analysis. There are also literatures illustrating the application of random walk on graph presented in [8], [9]. These literature has discussed the random walk profoundly from a specific aspect, but what they don't provide is the big picture of random walk applied in computer science society. In this paper, we will show the whole picture of random walk in the field of computer science.

When we talk about random walk in computer science community, there is one well-known algorithm that can't be omitted. It is is page rank algorithm [10]. Not only because of the great result it has achieved in practical problems, but also because it provides a objective way to measure the closeness between vertices. Based on the random walk, there are many other proximity measures. some of them are variants of page rank, such as HITS [11], SALSA [12], personalized page rank [13] and Simrank [14].

Based on these proximity measures, random walk plays an important role in all the areas of computer science. In the area of collaborative filtering, researchers have introduced some algorithms based on the proximity measures [15], [16]. There some other alternative approaches to solve the problem of collaborative filtering, but these approaches can't incorporate large of contextual information. Link prediction and recommender system are essentially the same as the collaborative filtering. They all aim to calculate the **k-most-close** vertices for one node. Hence the random walk view is also effective in these fields [17], [18], [19]. Random walk is also used in computer vision [20], [21], [22], [23] and semi-supervised learning [24], [25], [26], [27].

The rest of this paper is organized as follows. In section 2, we will introduction some basic concepts of random walk.

2. Brief Introduction of Random Walk

Random walk is an important part of stochastic process. Stochastic process can be denoted as $\{\xi_t, t = 0, 1, 2, ...\}$. ξ_t is a random variable. Single step transition probability can be denoted as $P\{\xi_{t+1} = j|\xi t = i\}$. T steps transition probability are defined as follows.

$$p_{ij}^{(t)} = P\{\xi_{s+t}|\xi_t = i\} \tag{1}$$

A graph is denoted by $G = (V, E)$, where V denotes the vertex set and E denoted the edge set. The adjacency matrix is denoted by A, where the A_{ij} means the weigh on edge i, j. The transition probability (single step) between node i and node j on the graph can be defined as follows.

$$p_{ij} = \frac{A_{ij}}{\sum_j A_{ij}} \tag{2}$$

We employ the diagonal matrix D to record for each node. In that case we can define the transition matrix of the graph as follows.

$$P_{ij} = A_{ij}/D_{ii} \tag{3}$$

P denotes the transition matrix of the graph. The Laplacian of G can be defined as follows.

$$L = D - A \tag{4}$$

3. Proximity Measures

In this section, we will discuss the proximity measures based on random walk. The proximity measures have been frequently used in the scope of computer science especially in the graph algorithm. we can roughly divide these measures into two classes, Pagerank and its variants, hitting time and commute time. Pagerank and its variants are originally presented to serve a search engine. Hence we can break it into query irrelevant proximity measures and query relevant proximity measures. The query irrelevant measures include pagerank and HITS, and the query relevant measures include personalized pagerank. Note that we would like to have a comprehensive way to get to know classical proximity measures, some other variants of pagerank such as SALSA,Simrank etc. will not be introduced in this section. And some other non-random-walk proximity measures like Katze score will also be omitted in this section. You can refer to these literature [14], [28], [29], [30], [31], [32] for further information. Since these proximity measures are so popular, there are many researchers devising fast algorithms to calculate these proximity measures [33], [34], [35], [36]. To focus on simplicity, we will not discuss about these fast algorithms a lot.

3.1. PageRank

The most famous Proximity measure is the pagerank. It was first proposed by Lary Page [10]. The purpose of this measure is to rank the webpage in World Wide Web. The network of webpage is considered as a graph where random walk happens. The graph is made up by vertexes and edges. The webpages are considered as the vertexes. If there is a webpage containing a hyperlink which pointing to anther webpage, then there should be a directed edge between these two vertexes. The direction of the edge is as same as the web redirection. The most simple page rank can be describe by the mathematic equation.

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} \tag{5}$$

$R(u)$ is the rank of web u. $B(u)$ is the set of vertexes point to page u. $out(u)$ is the set of pages u points to. $N(v)$ is the number of vertexes in set $out(u)$. The intuition behind this equation is that a page is important when it has more backlinks. The more important these backlinks are, higher rank this page gets. But this simple page rank cant be implemented because the practical situation is much more complicated. A more reliable mathematic description of page rank is as follows.

$$V = (1-\alpha)P^T V + \frac{\alpha}{n}\mathbf{1} \tag{6}$$

Alpha is the probability that the random walk restart in given steps. Alpha is crucial for this proximity measure. It makes sure the random walk procedure is aperiodic and irreducible. In that case, the random walk in the web network can converge to a certain distribution. However, the calculation of page rank uses a power method. To improve the converge speed of page rank, Extrapolation [37] present a novel algorithm for Page rank computation. Quadratic Extrapolation accelerate the convergence of the power method. The main strategy in this algorithm is periodically reducing estimates of the non-principal eigenvectors.

3.2. HITS

Pagerank has nothing to do with user-supplied query. Therefore, Jon M. Kleinberg [38] presents *Hyperlink-Induce topic search* which can filter the search result for a broad topic. The author proposes that there are two kinds of useful web-pages for topic search: authorities and hubs. He also proposes a link-base model for the conferral of authority. There are millions of pages relevant to a broad topic. Authorities are the most central pages for the broad topic, which provide good information for the broad topic. Whereas the hubs are those pages contains hyper-links redirecting to the authorities. Now, we can discuss that main procedure of how the author provide a broad topic search result for the user. Given a query, the author construct a focused subgraph of the *World Wide Web* relevant to the broad topic. The subgraph contains a set of relevant pages rich in candidate authorities. Then the author present an algorithm to discover authorities over the subgraph. The author would like to construct a subgraph denoted by S which satisfy the following requirements:

- S is relatively small.
- S is rich in relevant pages.
- S contains most the strongest authorities.

How to construct subgraph S is the most difficult problem in *HITS*. The author rendered a solution:

1) Collect the t highest ranked pages for the query as the root set R.
2) Expanding R along the links that enter and leave it.

Now we get a qualified subgraph. we can apply the algorithm over it. The author would like to extract these authorizes from the subgraph. Hence the author use two scores to describe a vertex in the subgraph: *authority score* and *hub score*. The intuition of this idea is that a node is good hub if it points to many authorities; a node is a good authority if it is pointed to by many good hubs. In order to break this circulation, the author use a iterative method, which can be mathematically describe as follows.

$$a(i) \leftarrow \sum_{j:j \in I(i)} h(j) \tag{7}$$

$$h(i) \leftarrow \sum_{j:j \in O(j)} a(j) \tag{8}$$

$I(i)$ denotes the set of pages point to page i. $O(i)$ denotes the set of pages pointed to by page i. $a(i)$ is the authority score of page i. $h(i)$ is the hub score of page i. During the iteration, authority score and hub score are normalized so their squares sum to 1. The two equation could be rewritten by using the matrix. A denote the unweighted adjacency matrix of the subgraph, vector a is the authority scores and vector h is the hub scores.

From the above equations, we can know that the hub scores converges to the principal eigenvector of AAt, meanwhile the authority scores converge to the principal eigenvector of AtA.

3.3. Personalized PageRank

Pagerank is a very democratic [39] since the walker can jump to every vertex with the Probability of alpha. On the contrary, personalized PageRank concentrate on one vertex. The intuitive idea of personalized page rank [40] is the random walker can jump to a certain vertex with the probability of alpha. The mathematical description is the following equation [7].

$$\mathbf{v} = (1-\alpha)P^T\mathbf{v} + \alpha\mathbf{r} \tag{9}$$

There are many link prediction problem using personalized page rank as a proximity measure [30]. We will discuss these applications of personalized pagerank in the following sections.

3.4. Hitting Time and Commute Time

Hitting time [41] can be considered as a weighted path length from i to j. The mathematic definition of hitting time is as follows:

Hij denotes the hitting time from node i to node j. P_{ik} denotes the transition probability from node I to node k. As we have mentioned before, on the undirected graph, transition probability matrix is symmetric. However, the hitting time matrix is not symmetric even on the undirected graph. Another important fact about hitting time was proved by Lovasz [8]: hitting time follows the triangle inequality. The commute time from node I to node j is defined as :

$$C_{ij} = h_{ij} + h_{ji} \tag{10}$$

In order to research the **commute time** on undirected graphs, Ashok K. Chandra et al. [42] give an electrical network view. They compare commute time between two nodes on graph to resistance on electrical network. They give us some intuition about commute time on undirected graphs:

- The smaller resistance can make the current go through easier on electrical networks, the smaller commute time can make random walker diffuse easier on undirected graphs.
- Commute time should be robust to small perturbation since removing or adding a few resistances do not change much on an electrical network.

4. Applications of Random Walk

Random walk have been successfully applied in different era of computer science such as social network analysis, computer vision and so on. Many applications of random walk are based on the proximity measures mentioned in the proximity measures section 3 such as recommender system, link prediction and collaborative filtering. These applications' intuition is to construct a graph or network for random walk. There are also some different ideas when researchers applying random walk on computer vision and semi-supervised learning.

4.1. Collaborative Filtering

Collaborative filtering is a method of making automatic predictions about the interests of a user by collecting preferences or taste information from many users. The assumption of collaborative filtering is that two people who have the same taste on one issue will have the same interest on the other issues.

Much literature has recorded methods of collaborative filtering with successful demonstrations of Bayesian, nonparametric, linear methods etc [43]. All these methods are essentially the same. They all match the individual to others based on their choices, and use combination of their experiences to predict future choices. However, Brand et al [16] introduced a random walk view to collaborative filtering. The goal of Brand is to find out what products a customer wants to buy next. He want to find out what product categories are preferred by a specific demographic group. They derived a weighted association graph from a relational database. These weighted association graphs include consumers and their web browsing behavior, shopping behavior and entertainment choices, etc. Figure 1 is a fragment of the association graph derived from the relational data base.

The authors look at the expected behavior of random walk on the association graph. Based on the hitting time and commute time, the authors employ a novel measure of similarity — the cosine correlation between states. Compared with other methods of collaborative filtering, one of the biggest advantages of random walks view is that it can incorporate large amounts of contextual information. By using cross-validation, the author proved that the random walk view collaborative filtering is more predictive and robust to perturbations of edges on the association graph

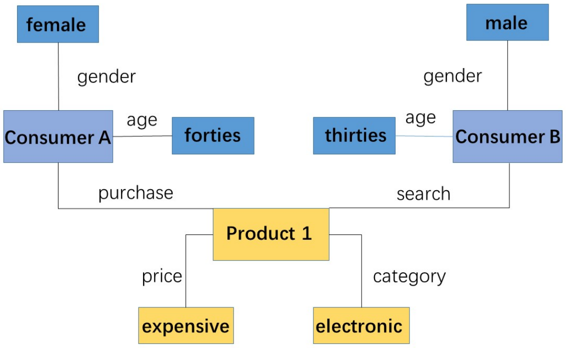

Figure 1. Association Graph

than other methods. The flaw of random walk view is the heavy computing price. In that case, the authors employ approximation strategies to alleviate time complexity.

Fouss et al. [15] also use random walk for movie collaborative recommendation. The authors also consider relational database as a collection of element sets linked by their connection. The authors exploit the graph structure of the relational database to compute dissimilarity measure between elements in sets. The dissimilarity is based on the hitting time and commute time. It was also the first time that hitting time and commute time measures were used in collaborative recommendation. For better understanding, the author gives us a specific example of the collaborative movie recommendation. If we get three elements, people, movie and movie category, relationships that between people and movie, and between movie and movie category. we have to do following things for movie recommendation:

- Compute dissimilarity measure between people based on the movies that they have watched.
- Compute dissimilarity measure between people and movies for recommendation.
- Compute dissimilarity measure between people and categories to give a prefer category for each person.

As a conclusion, Fouss et al. [15] introduce a general procedure for computing similarity between elements of a relational database. These elements are not necessarily directly connected. The authors use movie recommendation as an example to show us that their method has better performance than shortest path method on recommendation. However, there are two shortcomings of the authors' method. For large databases, this method is time consuming and does not scale well. The other short coming is that the method is valid on a weighted, undirected graph.

Although the random walk view of collaborative filtering is useful and has good performance, it also faces several challenges. One of the biggest problem is the start-up problem presented by Resnick [44]. All collaborative filtering systems are based on an existed database. If there isnt an existed database, the system cant be built up.

4.2. Link Prediction

Link prediction is used to predict the links that may exist in the future of evolving networks. Link prediction problem is a long-standing challenge in both computer science community and information science community. Random walk is one of the useful approaches to solve link prediction problem. Just like collaborative filtering, a random walk view of link prediction is also based on proximity measures.

Liben-Nowell et al. [30] present the performance of different proximity measures on link prediction problem such as hitting time and commute time, Katz score [29], SimRank etc. They test the different measures on the coauthership network of physics. They consider that the coauthor network is evolutionary with time going by. The coauthor network is denoted by $G =< V, E >$ in which each edge $e =< u, v >$ represents coauthorship between node u and node v that appear at time $t(e)$. They choose four time $t_0, t_0', t1, t_1'$ first. The four time has following relation: $t_0 < t_0' < t_1 < t_1'$. They apply algorithms in the network of $G[t_0, t_0']$ and output a list of edges that may appear on the network in the near future. And $G[t_1, t_1']$ are considered as the coauthorship network in the near future. They call $[t_0, t_0']$ as the training interval and $[t_1, t_1']$ as the test interval. In order to evaluate different algorithms, they use two parameters *ktraining* and *ktesting* to see how accurate the new edges between two vertices can be predicted. According their results, there is no winner among the different measures. But compared with the random predictor, many methods have much better performance, which indicates that the random walk view of link prediction works. They also find that the hitting time and commute time measures suffered from the information far away. The most effective proximity measure is the *Katz Score* which ensembles the paths between two nodes. Moreover, the computation of hitting time and commute time is time consuming.

To address this problem, Sarkar et al. [45] come up with the idea that to replace commute time with Truncated-commute-time in link prediction task. Based on the idea, they present a novel algorithm called GRANCH [45] to find out which two nodes will have an edge in the near future. The main intuition of GRANCH is that consider the graph as n overlapping subgraphs. Every subgraph is a bounded neighborhood for each node. And the hitting time is redefied as the random walk from any node in this neighborhood to the destination. They apply GRANCH in both simulated data and real world graphs. The authors empirically show that GRANCH reduces the computation and storage while retaining the performance of link prediction methods that based on commute time proximity measure. Link prediction also help researchers find out the potential relation between miRNAs and diseases [46]. They consider the miRNA-Disease heterogeneous network as two overlapping sub-networks: miRNA similarity sub-network and diseases similarity sub-network. They employ random walk with restart to predict the miRNA-disease associations in the heterogeneous network. This is a case that random work help the biologists to do their research more conveniently.

4.3. Recommender System

Recommender system with random walk approaches sometimes is just like the intuition of collaborative filtering. The inherent quality is just the same. They all exploit random walk to calculate the **closeness** between vertices. One of the persuasive evidences is that there is a literature [47] compare algorithm for recommender system with collaborative filtering algorithm presented in [15]. Hence we will give a specific application scenario to introduce the recommender system based on random walk. The scenario is recommending paper for researchers.

Some scholars use random walk to solve their own problem during doing research. They find that it is hard for them to find out useful literature recently published in their field. A researcher is supposed to be well aware of recent development of the field he is working on. So a paper recommendation system can help them find out potential helpful papers which is meaningful and time saving. Because publications increase exponentially, selecting useful papers is really a pain in the neck for the most researchers.

A very simplified algorithm to solve this problem is presented by Woodruff et al. [17]. The authors employ spreading activation and citation data to generate recommendations. The authors use documents read by the reader as input. and output the most related literature to the reader in a digital book. This method only recommends a chapter or an article in a digital book. A more applicable method for paper recommendation can be found in [18]. The author exploit collaborative filtering for recommendation. They use the citation web as the graph to create ratings. The author investigate six algorithms by do experiments on the subset of *ResearchIndex*. These algorithms can either provide relevant recommendations or novel recommendations, but none of them can do the both. And The use of citation web can affect the recommendations greatly.

Also based on the citation graph, Gori et al. [48] exploit the idea of page rank algorithm to solve the paper recommendation problem, and devise the *PaperRank* algorithm. The authors' view is that utilizing the model expressed by the citation graph can help us find out valuable papers to suggest to a user. The authors considered that the *PaperRank* algorithm must have two properties: propagation and attenuation. Propagation can help us find out a paper is a good suggestion for a researcher, if the paper is relevant to good papers in bibliography of the researcher's work. As for attenuation, it means that the positive influence of good papers decreases if we move further and further away from good papers on the citation graph. PageRank algorithm has both properties mentioned above. The author borrows its idea to solve paper recommendation problem. The essential of *PaperRank* algorithm is a random-walk-based score algorithm.

Xia et al. [49] incorporates author relations and historical preferences for scientific article recommendation. The authors build a graph based on the information of co-authors' relationships, and they employ the random walk with restart to generate a recommendation list. Compared with some baseline algorithms, the algorithm presented in the literature called CARE performs better in precision, recall, and F1 score. Most studies of paper recommendation use the algorithms that pay no attention to the different situation of researchers. But CARE method takes researchers own features into consideration. Hence the CARE method is more accurate than the baselines.

4.4. Computer Vision

Many researchers solve computer vision problems by using random walk. One of the popular techniques is characterizing shape of picture by using random walk. Gorelick et al. [50] compute many useful properties of a silhouette based on the notion of random walk. For every internal pixel in the contour, they compute a criterion reflecting the mean time required for a random walker beginning at the pixel to reach the boundary. Based on the computed value, they can extract many properties of the silhouette such as part structure, rough skeleton, local orientation, convex part, and concave part. Random walk is also utilized in image segmentation.

Meila et al. [20] present an approach of image clustering and segmentation based on the view of random walk proximity measures. They also find that the spectral view of clustering and segmentation have a probabilistic foundation. They exploit the eigenvalue and eigenvector of walkers transition matrix to cluster and segment image.

Grady et al. [51] propose a new algorithm for performing multi-label and interactive image segmentation. The interactive image segmentation means that the user has to label some pixels in the image manually. The algorithm can determine the probability of the random walker which starting from an unlabeled pixel reaching the predefined pixels. Therefore, a good segmentation of that image arises from the labels of all the pixels. The predefined labels indicate that the regions of the image belong to several objects. The authors treat the picture as a graph including nodes and edges. Nodes represents the pixels of the image. Edges represents the connection of two nodes, and the weigh of the edge means the likelihood of a random walker going through that edge. The authors believe that this view of image segmentation has two advantages: no discretization errors and no ambiguity. One reason of no discrestization is that the authors use purely combinational operators that require no discretization. The segmentation algorithm only requires the solution to a sparse, symmetric, positive definite system of equation, hence the efficiency of this algorithm is guaranteed.

Qiu et al. [23] exploit the properties of the commute time to develop image segmentation method. They compute the commute time from the spectrum. By using the discrete Greens function of graphs, they can analyze the cuts of the image from commute time. Qiu et al. also use commute time to motion track [52]. The main purpose of using commute time as proximity measure is to alleviate the effect of noise on the shape interaction matrix. The noise on the shape interaction matrix results in the loss of block-diagonal structure and the difficulty of the assignment of elements to objects. Commute time is a more robust measure than raw proximity matrix when facing the noise on the shape interaction matrix. The authors compute the commute time by using the Laplacian matrix shown in equation 4. They also show us that how the ensemble the commute time, kernel of PCA (Principle Component Analysis), the Laplacian eigenmap The function of commute time is to provide a proximity measure for the approaches provided by Qin in [23], [52].

4.5. Semi-supervised Learning

Semi-supervised learning uses both labeled data and unlabeled data for training. The goal is to classify the unlabeled data when the labeled data is just a small fraction of the dataset.

Zhu et al. [53] present a new approach of semi-supervised learning based on the random walk. They do classification task in continuous state space rather than in the discrete label set. The intuition of the approach is that the data points should be labeled as same as their neighbors. How to select the neighbors of a data point is a problem in front of us. The authors' strategy is to employ a harmonic function $f : V \rightarrow R$ on graph G. The harmonic function has a constrain on the labeled data i:

$$f(i) \equiv fl(i)y(i) \tag{11}$$

The harmonic function, which provides a consistent probabilistic semantics, is the basis of this semi-supervised classification approach. Since the author do classification in the continuous state space, they have to turn the continuous state space into discrete label set. Instead of employing a simple threshold in terms of the interpretation of random walk, the authors incorporate the prior knowledge by using *class mass normalization*(CMN) procedure. The promising result has shown that the approach can improve the accuracy of classification by exploiting the structure of unlabeled data.

Szummer et al. [54] the partially labeled data may be in the sub-manifold space, hence a measure of global similarity is needed for semi-supervised learning. Meanwhile, the authors also hope the measure can incorporate the structure of manifold. Based on these consideration, they present a Markov random walk model to classify the data. The research of [55], which shows how to change the distance matrix into a Markov process, helps a lot with the construction of graph. In that case, the representation of data set arises naturally. Given a partially labeled data set $\{(X_1, \tilde{y}_1), \cdots, (X_L, \tilde{y}_L), X_{L+1}, X_N\}$ in which L is much smaller than N, the authors represent the data set as a graph where node k represents the data $(X_k, \tilde{y}_k)$ or X_k. For node k, $P_{0|t}(i|k)$ denotes the probability of the random walker from node i to node k after t steps. They classify node k with the label c when c maximizes the following formula.

$$P_{post}(y|k) = \sum_i P(y|i)P_{0|t}(i|k) \tag{12}$$

$P(y|i)$ is an unknown parameter, which can be estimated by two techniques: maximum likelihood with Expectation Maximization(*EM*), and maximum margin subject to constraints. They discuss the two techniques in the paper and empirically show that the margin estimation has better performance. In a word, the authors provide a novel approach for semi-supervised learning task when the data sets with significant manifold structure. The parameter t in this approach is also important. T, denoting the number of transitions, determines the smoothness of random walk. However, the choice of t can be tricky and subjective. To overcome this little problem, Azran [26] presents the rendezvous algorithm. Just the same as the work of Szummer [54], the author represents the data points as nodes of a graph and employ the random walk view to do classification. The intuition of the authors' approach is the labels propagation over the graph. But the rendezvous algorithm is different. The labeled points dont propagate, but absorb the states of the random walk. The probability of each unlabeled data to be absorbed by different labeled points can be used to derive a distribution as the transition steps increase to infinity. Hence the rendezvous algorithm doesn't bother to choose a good value of the parameter t. The author draws a conclusion that the location of labeled point in the data set is as important as the size of labeled data set in terms of the experiments results.

5. Quantum View of Random Walk

The scalable quantum computer is a topical issue, hence the approaches of quantum computation and quantum information are popular topics nowadays. It is essential and inevitable that we introduce a quantum view of random walk. There are much literature that gives us explicit introduction to quantum random walk in a comprehensive way [6], [56], [57], [58]. In this section, we will just using a simple example of one dimensional quantum walk to give you a brief introduction, and focus more on the application of quantum walk in computer science society.

Kempe et al [56] presented us two kinds of quantum walks including discrete time quantum walk and continuous time quantum walk. We will give an easy one-dimension example to help quickly know the basic idea of discrete times quantum walk and continuous time quantum walk.

5.1. Discrete Time Quantum Walk

The discrete time model first appeared in the work of Feynman [59] in 1966. In the field of quantum computation, Meyer rediscovered the discrete time model of quantum walk [60] [61]. We define a space $H = H_p \bigotimes H_c$ for one dimensional quantum walk, . H_p denotes Hilbert space.

For one dimensional Hilbert space, it can be represented as follows.

$$H_p = \{|i\rangle : i \in Z\} \tag{13}$$

H_c is spanned by two basic states $\{|\uparrow\rangle, |\downarrow\rangle\}$. Operation S defines the translation on space H.

$$S = |\uparrow\rangle\langle\uparrow| \otimes \sum_i |i+1\rangle\langle i| + |\downarrow\rangle\langle\downarrow| \otimes \sum_i |i-1\rangle\langle i| \tag{14}$$

S can transform the basic state $|\uparrow\rangle \otimes |i\rangle$ to $|\uparrow\rangle \otimes |i+1\rangle$ and $|\downarrow\rangle \otimes |i\rangle to| \downarrow\rangle \otimes |i-1\rangle$.

C is a unitary transformation to rotate the spin in H_c. A frequently used unitary transformation is called Hadamard coin H. Here is an example of H.

$$|\uparrow\rangle \otimes |0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \otimes |0\rangle \tag{15}$$

The single quantum walk transformation can be defined as follows.

$$U = S \cdot (C \otimes I) \tag{16}$$

Here is a example of single step transformation.

$$|\uparrow\rangle \otimes |0\rangle \xrightarrow{U} \frac{1}{\sqrt{2}}(|\uparrow\rangle \otimes |1\rangle + |\downarrow\rangle \otimes |-1\rangle) \tag{17}$$

The T steps of transformation can be represented by U^T.

5.2. Continuous Time Quantum Walk

The original purpose of continuous time quantum walk is to speed up many a algorithm using classic random walks. The concept of continuous time quantum walk was first presented by Farhi et al. in 1997 [62]. The authors exploit quantum walk in the decision tree algorithm instead of classic random walk. Different from discrete time quantum walk, continuous time quantum walk don't need a coin space Hc, taking place entirely in the Hilbert space Hp. [56] . The idea of continuous time quantum walk is from continuous random walk. The continuous time random walk can be defined as

$$P(t) = exp(-Ht)P(0) \tag{18}$$

Similarly, the unitary time evolution operator of continuous time quantum walk is

$$\hat{U}(t) = exp(-i\hat{H}t) \tag{19}$$

5.3. Algorithms Based on Quantum Walk

In this section, we are going to introduce some algorithms based on the two quantum walk model mentioned above on solving practical problems. We can find some different properties between quantum walk and classical random walk through these examples. To help understand quantum walk algorithms better, we would like to separate the algorithms into two categories depending on the model they use. The first category is the continuous time quantum walk algorithm—quantum decision tree algorithm. The other category is based on discrete time quantum walk algorithms, including quantum page rank algorithm and element distinctness algorithm.

5.3.1. Quantum Decision Tree Algorithm.

Fahri [62] originally presented the idea of continuous time quantum walk with the example of decision tree algorithm. He chooses the approach that systematically exploring the whole tree with a probabilistic rule. The author aims to achieve that the n-level nodes can be reach in polynomial time with a considerable probability. A tree is a penetrable tree when its node or nodes in n-level meet the requirement above. If a tree is penetrable for an specific algorithm, we believe that the problem corresponding to the decision tree is solvable with this algorithm in the polynomial time. The author presents the quantum walk algorithm for decision tree with following intuition. He considers decision tree nodes as quantum states in *Hilbert* space. Then he constructs a Hamiltonian $\hat{H}$ which determines the time evolution of the quantum system. With the basis of Hamiltonian, the author presents the unitary time evolution operator shown in the Equation 19. The author compare quantum walk decision tree algorithm with classical counterpart, and find that there is a family of trees which are both classical penetrable and quantum penetrable. However, Some decision trees is quantum penetrable but not classical penetrable. With these findings, the author draws the conclusion that quantum algorithms are more faster with respect to the classical decision tree algorithms for some decision tree problems.

5.3.2. Quantum Page Rank Algorithm.

page rank algorithm is one of the most important random walk algorithms. When quantum computation is widely considered in the era of random walk, it is natural and inevitable to apply quantum computation on page rank algorithm. There is much literature of quantum networks [63], [64], [65], [66], [67], [68]. In order to study the behavior of pagerank algorithm in the quantum network, the authors present the quantum page rank algorithm [69]. However, the authors don't give a specific defination of quantum page rank algorithms, but give a admissible class shown as follows.

1) The classical PageRank must be embedded into the quantum class with its undirected graph structure preserved.
2) The sum of all quantum PageRanks must equal to 1.
3) The quantum PageRank obeys a quantized Markov Chain (MC) rules.
4) The classical algorithm to compute the quantum PageRank is also feasible.

The author exploit the idea of discrete time quantum walk. Hence we have to define the coin space H_c and Hilbert space H_p which are mentioned in the section of discrete time quantum walk. The definition of coin space is similar to the one dimension quantum walk.

$$H_c = span\{|L\rangle, |R\rangle\} \tag{20}$$

However the Hilbert space $H_p$here is a little tricky. Since the page rank algorithm is on a graph, the author define the Hilbert space as the space of oriented edges.

$$H_p = span\{|i\rangle_1, |j\rangle_2 \quad | \quad i, j \in N\} \tag{21}$$

where N denotes the all the vertices of the graph. Since the edge is oriented, We use the subscript 1,2 to show the direction.

With these defination and the method of Szegedys Quantization of Markov Chains [70]. We can present the unitatry step operator of quantum walk as follows.

$$U = S(2\Pi - \mathit{1}) \tag{22}$$

$$S = \sum_{i,k=1}^{N} |j,k\rangle\langle k,j| \tag{23}$$

$$\Pi = \sum_{j=1}^{N} |\psi_j\rangle\langle\psi_j| \tag{24}$$

$$|\psi_j\rangle = |j\rangle_1 \otimes \sum_{k=1}^{N} \sqrt{G_{kj}}|k\rangle_2 \tag{25}$$

Where G_{ij} means the weight of edge ij.

The authors apply quantum pagerank algorithm on small generated network to have a insight of the behavior of it. They find that the quantum pagerank algorithm obtain a larger score than the classical value. In the meanwhile, the quantum algorithm break down the hierarchy of classical values. The authors also look into the properties of quantum page rank algorithm in complex real-world networks [71]. The authors find that quantum page rank algorithm can reveal the underlying topology of the network more univocally with respect to classical page rank algorithm. The ability of detecting hub for network is enhanced with respect to classical counterpart.

5.3.3. Element Distinctness.

We introduce the element distinctness problem first. Element distinctness problem is to tell whether all the elements in a given sequence are distinct. More precisely, $M = x_i, i \in N$, are there $x_i \in Mandx_jinM$ and $i \neq j$ such that $x_i = x_j$? There is a simple classical algorithm to solve this problem with $Nlog(N) + O(N)$ comparisons. Buhrman et al. present an quantum algorithm to speedup [72]. Their algorithm give a upper bound of computation cost $O(N^{\frac{3}{4}}log(N))$. Ambainis [73] improve the quantum way to solve element distinctness with $O(N^{\frac{2}{3}})$ comparisons. The intuition of this optimal quantum algorithm is to construct a graph, and transform the element distinctness problem of finding a marked vertex in the graph. In order to search marked vertex efficiently, the author improve the Grover's quantum search algorithm [74], [75]. The author reuses the information that queries before, and search a marked vertex with $O(N^{\frac{2}{3}})$ comparisons instead of $O(N)$ comparisons in Grover's search algorithm.

6. Conclusion

In this paper, we introduce the random walk from the view of application in computer science. We get to know some prerequisite knowledge of understanding proximity measures based on the random walk first. Then we discuss about some classical proximity measures in a comprehensible way. Some complex variants of proximity measures we don't pay much attention to, but they won't be obstacles for us to have a whole picture of proximity measures based on random walk in our mind. With a map of proximity measures, we can find a clear path through the forest of random walk algorithms. We talk about proximity based algorithms like link prediction algorithms, collaborative filtering algorithms, and recommender systems. There are also some machine learning problems that can be well solved from the perspective of random walk. In the field of computer vision, random walk help to solve the problem of graph segmentation. In semi-supervised learning, random walk is also a effective approach. With the development of quantum computation and quantum information, it is necessary to introduce a quantum view of random walk. In case we are lost in the tricky concepts of quantum quantum mechanism, we first introduce discrete-time quantum walk and continuous-time quantum walk with simple one dimensional examples. Then we have discuss about the classical applications of quantum walks. It is exciting to find that there are many different properties and behaviours between random walk and quantum walk. We also introduce some examples that quantum walk approach can gain a remarkable speedup. There are also an important part that we missed. We don't look into the performance of random walk in practical problems.

7. Open Issues

We are in the era of information explosion. We produce so many data every day. Hence the real-world network is so giant. When we apply random walk on the giant complex real-word network, there are two challenges in front of us. The first one is the speed of the random walk algorithms. The second one is the main-memory volume.

7.1. Speed of Random Walk Algorithms

The time complexity of graph random walk kernel is at least $O(n^3)$ or $O(m^2)$ for graph with n nodes and m edges. [76] In an artificially generated graph, this time complexity is acceptable, but in a giant graph it is a disaster. There are already researchers coping with this issue. Kang et al. [76] propose ARK graph kernels with time complexity $O(n^2)$ or $O(m)$. There is a prerequisite for this graph kernel. The graph must have lower intrinsic ranks than the order of the graph. Tong et al. also realize the speed problem in random walk with restart [77]. The algorithm with restart needs cubic pre-computation time. The authors also exploit the low-rank property to make random walk with restart faster.

7.2. Problem of main-memory Volume

All the fast random walk graph kernels or algorithms are under the consumption that the whole graph can be fit in the main-memory. But with the real-world networks or graph becoming larger and larger, this condition can't be satisfied any more. One of the solutions is to divide the graph into several clusters. There are literatures provide some approaches for graph partition and clustering on giant network. [78], [79] One of the most popular method is METIS [79]. Since more and more researchers pay attention to the giant network problem, there are a more effective clustering algorithm for graph clustering and a better method to apply random walk on giant network with external memory [80]. The author call the clustering method RWDISK. RWDISK has been proved to be a better way for graph partition on several famous datasets such as DBLP, citeseer and so on. But these method still has a unacceptable time latency with respect to enormous graph.

Acknowledgments

The authors would like to thank...

References

[1] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, no. 1865, p. 294, 1905.

[2] F. Spitzer, "Principles of random walk. (zz)," *Springer Verlag Gmbh*, 1976.

[3] J. Rudnick and G. Gaspari, "Elements of the random walk," *Contemporary Physics*, vol. 46, no. 2, pp. 147–148, 2004.

[4] G. H. Weiss, "Aspects and applications of the random walk," 1994.

[5] B. D. Hughes, "Random walks and random environments, vol. 1: Random walks," *Biometrics*, vol. 54, no. 3, pp. 1675–1677, 1998.

[6] S. Venegas-Andraca, "Quantum walks: A comprehensive review," *Quantum Information Processing*, vol. 11, no. 5, pp. 1015–1106, 2012.

[7] P. Sarkar and A. W. Moore, *Random Walks in Social Networks and their Applications: A Survey*. Springer US, 2011.

[8] L. Lovsz, L. Lov, and O. P. Erdos, "Random walks on graphs: A survey," *Combinatorics*, vol. 8, no. 4, pp. 1–46, 1993.

[9] D. J. Aldous and J. Fill, "Reversible markov chains and random walks on graphs," *Journal of Theoretical Probability*, vol. 2, no. 1, pp. 91–100, 1999.

[10] L. Page, "The pagerank citation ranking : Bringing order to the web," *Stanford Digital Libraries Working Paper*, vol. 9, no. 1, pp. 1–14, 1998.

[11] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the Acm*, vol. 46, no. 5, pp. 604–632, 1999.

[12] R. Lempel and S. Moran, "Salsa:the stochastic approach for link-structure analysis," *Acm Transactions on Information Systems*, vol. 19, no. 2, pp. 131–160, 2001.

[13] D. Fogaras, B. Rcz, K. Csalogny, and T. Sarls, "Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments," *Internet Mathematics*, vol. 2, no. 3, pp. 333–358, 2005.

[14] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 538–543.

[15] F. Fouss, A. Pirotte, and M. Saerens, "A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation," *web intelligence*, pp. 550–556, 2005.

[16] M. Brand, "A random walks perspective on maximizing satisfaction and profit." pp. 12–19, 2005.

[17] A. Woodruff, R. Gossweiler, J. E. Pitkow, E. H. Chi, and S. K. Card, "Enhancing a digital book with a reading recommender," pp. 153–160, 2000.

[18] S. M. Mcnee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl, "On the recommending of citations for research papers," *conference on computer supported cooperative work*, pp. 116–125, 2002.

[19] M. Gori and A. Pucci, "Research paper recommender systems: A random-walk based approach," *web intelligence*, pp. 778–781, 2006.

[20] M. Meila and J. Shi, "A random walks view of spectral segmentation. aistats," *Ai & Statistics*, 2001.

[21] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[22] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, Nov 2006.

[23] H. Qiu and E. R. Hancock, "Image segmentation using commute times." 2005.

[24] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," pp. 912–919, 2003.

[25] M. Szummer and T. S. Jaakkola, "Partially labeled classification with markov random walks," vol. 14, pp. 945–952, 2002.

[26] A. Azran, "The rendezvous algorithm: multiclass semi-supervised learning with markov random walks," pp. 49–56, 2007.

[27] N. Tishby and N. Slonim, "Data clustering by markovian relaxation and the information bottleneck method," pp. 640–646, 2001.

[28] E. Adar, Lada A. Adamic, "Friends and neighbors on the web," in *Social Networks*, 2003, pp. 211–230.

[29] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[30] D. Liben-Nowell and J. Kleinberg, *The link-prediction problem for social networks*. John Wiley & Sons, Inc., 2007.

[31] J. Crnic, *Introduction to Modern Information Retrieval*. McGraw-Hill,, 1983.

[32] A. Broder, "On the resemblance and containment of documents," in *Compression and Complexity of Sequences 1997. Proceedings*, 2002, p. 21.

[33] B. H. Bloom, *Space/time trade-offs in hash coding with allowable errors*. ACM, 1970.

[34] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," *Journal of Computer & System Sciences*, vol. 60, no. 3, pp. 630–659, 2000.

[35] M. Najork, S. Gollapudi, and R. Panigrahy, "Less is more: sampling the neighborhood graph makes salsa better and faster," in *ACM International Conference on Web Search and Data Mining*, 2009, pp. 242–251.

[36] M. Najork and N. Craswell, "Efficient and effective link analysis with precomputed salsa maps," in *ACM Conference on Information and Knowledge Management*, 2008, pp. 53–62.

[37] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, "Extrapolation methods for accelerating pagerank computations," in *International Conference on World Wide Web*, 2003, pp. 261–270.

[38] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," in *SODA '98 : Proceedings of the Ninth Acm-Siam Symposium on Discrete Algorithms, Philadelphia, Pa, Usa*, 1998, pp. 604–632.

[39] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2011.

[40] T. H. Haveliwala, *Topic-sensitive PageRank: A context-sensitive ranking algorithm for Web search*. IEEE Educational Activities Department, 2003.

[41] D. Aldous and J. A. Fill, "Reversible markov chains and random walks on graphs-chapter 9: A second look at general markov chains," 2002.

[42] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks:*. Mathematical Association of America, 1984.

[43] G. Adomavicius and A. Tuzhilin, "Recommendation technologies: Survey of current methods and possible extensions," 2008.

[44] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens:an open architecture for collaborative filtering of netnews," in *ACM Conference on Computer Supported Cooperative Work*, 1994, pp. 175–186.

[45] P. Sarkar and A. Moore, "A tractable approach to finding closest truncated-commute-time neighbors in large graphs," *Proc Uai*, 2012.

[46] Y. Liu, X. Zeng, Z. He, and Z. Quan, "Inferring microrna-disease associations by random walk on a heterogeneous network with multiple data sources," *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, vol. PP, no. 99, pp. 1–1, 2017.

[47] M. Gori and A. Pucci, "Itemrank: a random-walk based scoring algorithm for recommender engines," in *International Joint Conference on Artifical Intelligence*, 2007, pp. 2766–2771.

[48] ——, "Research paper recommender systems: A random-walk based approach," in *Ieee/wic/acm International Conference on Web Intelligence*, 2007, pp. 778–781.

[49] F. Xia, H. Liu, I. Lee, and L. Cao, "Scientific article recommendation: Exploiting common author relations and historical preferences," *IEEE Transactions on Big Data*, vol. 2, no. 2, pp. 101–112, 2016.

[50] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the poisson equation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 28, no. 12, pp. 1991–2005, 2006.

[51] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[52] H. Qiu and E. R. Hancock, *Robust Multi-body Motion Tracking Using Commute Time Clustering*. Springer Berlin Heidelberg, 2006.

[53] X. Zhu, "Semi-supervised learning using gaussian fields and harmonic functions," *Proc Icml*, pp. 912–919, 2003.

[54] M. Szummer and T. Jaakkola, "Partially labeled classification with markov random walks," in *International Conference on Neural Information Processing Systems: Natural and Synthetic*, 2001, pp. 945–952.

[55] N. Tishby and N. Slonim, "Data clustering by markovian relaxation and the information bottleneck method," in *International Conference on Neural Information Processing Systems*, 2000, pp. 368–377.

[56] J. Kempe, "Quantum random walks: an introductory overview," *Contemporary Physics*, vol. 50, no. 1, pp. 339–359, 2003.

[57] T. A. Brun, H. A. Carteret, and A. Ambainis, "Quantum to classical transition for random walks," *Physical Review Letters*, vol. 91, no. 13, p. 130602, 2002.

[58] Y. Aharonov, L. Davidovich, and N. Zagury, "Quantum random walks," *Physical Review A*, vol. 48, no. 2, p. 1687, 1993.

[59] R. P. Feynman and A. R. Hibbs, "Quantum mechanics and path integrals (international series in pure and applied physics)," *Students Quarterly Journal*, vol. 37, no. 145, 1965.

[60] D. A. Meyer, "From quantum cellular automata to quantum lattice gases," *Journal of Statistical Physics*, vol. 85, no. 5-6, pp. 551–574, 1996.

[61] ——, "On the absence of homogeneous scalar unitary cellular automata," *Physics Letters A*, vol. 223, no. 5, pp. 337–340, 1996.

[62] E. Farhi and S. Gutmann, "Quantum computation and decision trees," *Phys.rev.a*, vol. 58, no. 2, pp. 915–928, 1997.

[63] C. Elliott, "The darpa quantum network," 2004.

[64] M. Peev, T. Lnger, T. Lornser, A. Happe, O. Maurhart, A. Poppe, and T. Themel, "The secoqc quantum-key-distribution network in vienna," in *Optical Fiber Communication - incudes post deadline papers, 2009. OFC 2009. Conference on*, 2009, p. OThL2.

[65] M. Fujiwara, H. Ishizuka, S. Miki, and T. Yamashita, "Field demonstration of quantum key distribution in the tokyo qkd network," in *Quantum Electronics Conference & Lasers and Electro-Optics*, 2011, pp. 507–509.

[66] D. Stucki, M. Legre, F. Buntschu, B. Clausen, N. Felber, N. Gisin, L. Henzen, P. Junod, G. Litzistorf, and P. Monbaron, "Long term performance of the swissquantum quantum key distribution network in a field environment," *New Journal of Physics*, vol. 13, no. 12, pp. 123 001–123 018(18), 2011.

[67] D. Lancho, J. Martinez, D. Elkouss, M. Soto, and V. Martin, "Qkd in standard optical telecommunications networks," in *International Conference on Quantum Comunication and Quantum Networking*, 2009, pp. 142–149.

[68] T. Langer and G. Lenhart, "Standardization of quantum key distribution and the etsi standardization initiative isg-qkd," *New Journal of Physics*, vol. 11, no. 5, p. 055051, 2009.

[69] G. D. Paparo and M. A. Martindelgado, "Google in a quantum network," *Scientific Reports*, vol. 2, no. 1, pp. 444–444, 2012.

[70] P. Richter and M. Szegedy, *Quantization of Markov Chains*. Springer US, 2008.

[71] G. D. Paparo, M. Mueller, F. Comellas, and M. A. Martindelgado, "Quantum google in a complex network," *Scientific Reports*, vol. 3, no. 1, pp. 2773–2773, 2013.

[72] H. Buhrman, C. Durr, M. Heiligman, P. Hoyer, F. Magniez, M. Santha, and R. De Wolf, "Quantum algorithms for element distinctness," *Journal of Applied Physics*, 2000.

[73] A. Ambainis, "Quantum walk algorithm for element distinctness," in *IEEE Symposium on Foundations of Computer Science*, 2004, pp. 22–31.

[74] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *arXiv: Quantum Physics*, 2000.

[75] L. K. Grover, "A fast quantum mechanical algorithm for database search," *symposium on the theory of computing*, pp. 212–219, 1996.

[76] U. Kang, H. Tong, and J. Sun, "Fast random walk graph kernel," 2012.

[77] H. Tong, C. Faloutsos, and J. Y. Pan, "Fast random walk with restart and its applications," in *International Conference on Data Mining*, 2006, pp. 613–622.

[78] G. Karypis and V. Kumar, *A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs*. Society for Industrial and Applied Mathematics, 1998.

[79] ——, "Metis: a software package for partitioning unstructured graphs," *International Cryogenics Monograph*, pp. pgs. 121–124, 1998.

[80] P. Sarkar and A. W. Moore, "Fast nearest-neighbor search in disk-resident graphs," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 513–522.