Predictive and Personalized Drug Query System

Samamon Khemmarat and Lixin Gao

***Abstract*—Several factors need to be carefully considered in using pharmaceutical drugs, such as drug interactions, side effects, and contraindications. To further complicate the matter, the presence of some drug properties, such as side effects, depends on patient characteristics, such as age, gender, and genetic profiles. Our goal is to provide a tool to assist medical professionals and drug consumers in choosing and finding drugs that suit their needs. We develop an approach that allows querying for drugs that satisfy a set of conditions. The approach allows users to specify patient profiles and tailors the answers to the profiles. We utilize drug data from multiple data sources. However, drug data are usually noisy and incomplete as they are either manually curated or automatically extracted from text resources such as drug labels. To cope with incomplete and noisy data, in contrast to traditional query systems, our approach considers both the answers that exactly match the query and those that closely match the query. We represent drug information as a heterogeneous graph and model answering a query as a subgraph matching problem. To rank answers, our approach leverages the structure and the heterogeneity of the drug graph to quantify the likelihood of edges and score the answers. Our evaluation shows that for quantifying the edge likelihood, our network-based approach can improve the AUROC (Area under Receiver Operating Characteristic) by up to 18%, comparing to a baseline approach. We develop a prototype of our system and demonstrate its benefits through several examples.**

***Index Terms*—query system, drug query, personalization**

I. Introduction

SEVERAL precautions should be taken in using pharmaceutical drugs, for both healthcare professionals, who prescribe and administer drugs, and for drug consumers. Factors such as interactions among the prescribed drugs, interactions with the patient's current medication, side effects to be avoided, and contraindications, need to be carefully considered. Additionally, the presence of some drug properties, such as side effects and effectiveness, depends on characteristics of patients, such as age, gender, lifestyles, and genetic profiles. Having to consider all these complicated factors can be a huge burden to professionals and drug consumers.

In this work, our goal is to provide a tool to assist professionals and consumers in finding and choosing drugs. To achieve this goal, we develop an approach that allows a user to query for drugs that satisfy a set of conditions based on drug properties, such as drug indications, side effects, and drug interactions. For example, for a patient whose occupation is a driver, a doctor may want to issue a query: *Find a drug for fever and allergy that does not cause drowsiness.* Supposing that a patient is currently taking some medicines, Enoxaparin and Aspirin, a query can be: *Find a drug for diabetes and a drug for epilepsy that do not interact with Enoxaparin and Aspirin and do not interact with each other*. Furthermore, the approach allows users to specify a patient profile and tailors the answers according to the profiles. For example, to find a schizophrenia drug for an elder female patient who has a heart disease, a query can be: *Find a drug for schizophrenia without the side effect of heart failure for a female patient, age 60.*

In order to answer drug queries, it is important that we have comprehensive drug information. Currently, there are several drug information sources that are open to public, such as DrugBank [2], SIDER2 [3], and KEGG Drug [4]. As these data sources offer different facets of drug information and have different coverage of drugs, we can combine data from these sources into a unified drug information base. However, it is non-trivial to use these data to answer drug queries effectively. Many of the data sources are manually curated and therefore have limited coverage. Some data sources contain information automatically extracted from text resources, such as drug labels and published articles, which may be prone to errors. Furthermore, many properties of drugs may not have been discovered. For example, long-time side effects of drugs may not be observed during clinical studies. Our challenge is to be able to provide good answers to the queries despite the incompleteness and the noisiness of data.

Considering the characteristics of drug data, traditional query systems that provide only the answers that exactly match the queries have several disadvantages. First, these systems can miss some answers that in fact can satisfy a query but do not exactly match the query due to the imperfection of the data. Second, by disregarding the possibility of incomplete information, the answers returned can be misleading. For example, if a query asks for a drug that does not interact with a particular drug, some of the drugs that interact with the given drug may also be given as answers because their interaction data is incomplete. Third, in the case that there are indeed no drugs that can satisfy a query, these systems do not return any answer to users, while offering answers that almost or partly satisfy the query could be helpful for the users to make further decisions.

To address the aforementioned problems, we propose an approach that considers not only the answers that exactly match the query but also those that closely match the query. To provide both exact and approximate answers, we model the problem of answering query as a subgraph matching problem, in which drug information is represented as a heterogeneous graph and a query is represented as a query graph. Then, we propose a score function for evaluating the quality of answers based on how well they match with a query graph. The score function approximates the closeness between two nodes, which corresponds to the likelihood that there would be

S. Khemmarat and L. Gao are with the Department of Electrical and Computer Engineering, University of Massachusetts Amherst, Amherst, MA, 01003. E-mail: {skhemmar,lgao}@ecs.umass.edu

This paper is a revised and extended version of the one that appeared in the 9th EAI International Conference on Pervasive Computing Technologies for Healthcare, 2015 [1].

an association between a drug and a drug property, by using a novel network-based approach. As a result, the incompleteness of data is explicitly taken into account.

The key contributions of this paper are as follows.

- We propose an approach for answering drug queries, which finds a drug or a set of drugs that satisfy conditions based on drug properties. The approach considers both exact- and close-match answers to provide useful answers despite incomplete and noisy underlying datasets. Additionally, the answers are personalized based on a given patient profile.
- We propose a score function for ranking the answers. The score function estimates the likelihood of association between drugs and drug properties and integrates the likelihood to score an answer.
- We propose an approach that leverages the network structure and the heterogeneity of the drug information graph to quantify edge likelihood in the graph. Our evaluation shows that the network-based approach outperforms a state-of-the art approach that relies on fine-grained drug properties, with up to 18% increase of overall AUROC (Area under Receiver Operating Characteristic) [5] and up to 37% increase of AUROC for drug properties with small positive training data.
- We develop a prototype of the drug query system, integrating data from DrugBank [2], SIDER [3], KEGG Drug [4], and FDA (U.S. Food and Drug Administration) drug adverse event reports. We demonstrate the benefits provided by our system through several example queries.

The rest of this paper is organized as follows. We discuss related work in Section II. Section III provides our problem definition. Our methodology is presented in Section IV. In Section V, we describe how our approach personalizes answers for specific patient profiles. Section VI describes our prototype system. Section VII presents the evaluation of our approach. We summarize our work in Section IX.

II. Related Work

There are multiple drug information databases available for public access, such as DrugBank [2], KEGG DRUG [4], and PharmGKB [6]. Our work leverages these databases in order to provide a decision support tool for medical practitioners and drug consumers. A few studies aim to answer medical questions, including drug-related questions, and provide decision support on drug prescription [7]–[10]. These works model drug information as a relational database or an RDF (Resource Description Framework) knowledge base. Their focus is on how to convert raw data into the knowledge base and how to translate questions in natural languages to SQL queries or SPARQL, a query language for RDF, in order to retrieve the answers.

Clinical decision support systems that assist physicians in drug prescription have been proposed. These systems can provide alerts on potential drug adverse events, such as drug interactions and drug allergy, and recommendations of drugs and dosing based on clinical data and available evidence [11], [12]. Our work is orthogonal to the aforementioned existing works. We focus on providing high quality and useful answers to users despite the incompleteness and noisiness of available data. Our approach can be used together with the techniques proposed in existing works to achieve a complete drug query system.

Several works have proposed techniques for answering queries over noisy data [13]–[16]. In these works, data is represented as a graph and answering queries is equivalent to finding subgraphs that match a given query graph. To handle data noisiness, for a given subgraph query, they return both similar and exact matches based on a similarity measure. Inspired by these techniques, our method represents drug information as a graph and defines a similarity measure specifically for drug queries, which incorporates the likelihood of associations between drugs and drug properties.

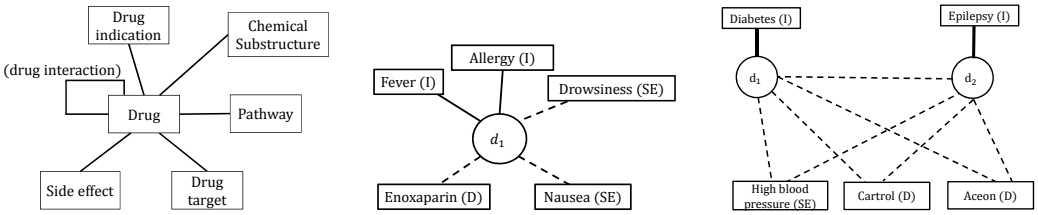
There are recent works on predicting novel drug properties (including drug targets, indications, and adverse effects), which are related to our problem of quantifying the likelihood of associations between drugs and drug properties. Methods that predict drug properties based on chemical structures [17]–[20] have been proposed. Some approaches analyze the correlation between drug targets, their corresponding biological pathways, adverse effects, and indications to perform prediction [21], [22]. These existing works suggest that various types of data are potentially useful in predicting drug properties. Network-based approaches have been proposed to discover novel drug-target interactions [23]–[25], drug-drug interactions [26]–[28], and drug adverse reactions [29]. In these approaches, networks containing drugs and drug property entities are created, and the network features, such as common neighbors or the number of paths, are used to predict drug properties. These existing studies have demonstrated that network structures can be used to effectively predict drug properties. However, in these approaches, the networks usually contain limited types of drug property entities. In our work, the likelihood of drug properties is quantified based on the structure of a heterogeneous drug information network, which contains various types of drug property entities, including targets, pathways, side effects, indications, and drug interactions. Our approach is also extensible to include other types of drug properties as more information is available.

III. Problem Description

We represent drug information aggregated from multiple data sources as a heterogeneous graph, i.e., a graph that has nodes of multiple types. Using the drug graph as a basis, we model the problem of answering queries as a subgraph matching problem. In this section, we describe the schema of the drug graph, the query graph, and our problem.

A. Drug Graph Schema

Drug information is represented by a graph $G(V_G, E_G, type_G, key_G)$, where V_G is a set of nodes and E_G is a set of edges. Each node in the graph represents an entity of a specific type, which is either a *drug* (D) or a type of drug properties, such as a pathway associated with a drug (P), a target protein of a drug (T), an indication of



(a) Schema of drug graph (b) Example query graph 1 (c) Example query graph 2

Fig. 1. Schema of the drug graph and example query graphs. (solid edges: positive, dashed edges: negative)

a drug (I), a side effect of a drug (SE), and a chemical substructure of a drug (CH). $type_G$ is a function that maps a node to a node type. key_G is a function that maps a node to a set of keywords related to the nodes, including identifiers and all the synonyms describing the node. For example, a node v that represents a drug has $type_G(v)$ = "Drug", and $key_G(v)$ includes the drug's generic names and brand names. Note that since each node has a specific type, a symptom (e.g., fever), which can be either a side effect or an indication, is represented as two distinct nodes in the drug graph.

The schema of the drug graph is shown in Figure 1a. An edge from a drug to a *drug property node* (a node of type P, T, I, SE, or CH) represents the fact that the drug has or is associated with the particular property. An edge between two drug nodes represents the fact that the two drugs can interact. Notice that the drug graph includes not only the drug properties expected to be in the queries but also drug chemical/biological information, such as chemical structures, targets, and pathways. These nodes allow us to establish relationships among drugs, which are useful for inferring potential associations between drugs and drug properties.

B. Query Expression

Now we describe queries on the drug graph. A query is represented as a graph $Q(V_Q, E_Q, type_Q, key_Q)$. We refer to nodes in the query graph as *query nodes*. The query graph follows the same schema as the drug graph, but in contrast to the drug graph, some nodes may not be assigned a keyword as they represent the information the user is looking for. Accordingly, we can divide the query nodes into two groups. (1) **Variable node**: A variable node represents the information the user wants to find. The keywords of the variable nodes are not given in the query. (2) **Reference node**: A reference node serves as a reference for identifying the variable nodes. Each reference node has a keyword specified by a user.

Each edge in the query graph has a **sign**, which can either be positive or negative. A positive edge means that the user wants the connection to exist between the two nodes, while a negative edge means there should be no connection between the two nodes.

We show two example query graphs in Figure 1. In Figure 1b, the query graph corresponds to the query: *Find a **drug** for **fever** and **allergy** that does not interact with **Enoxaparin** and does not cause **drowsiness** and **nausea***. In this query graph, there is one variable node, d_1, representing the drug to find. The drug d_1 is connected to the two indications with positive edges. To avoid side effects and interactions, negative edges are used to connect d_1 to the side effect nodes and another drug node. The query graph in Figure 1c is for the query: *Find a **drug** for **diabetes** and a **drug** for **epilepsy** that do not interact with **Cartrol** and **Aceon**, do not interact with each other, and do not cause **high blood pressure***. Notice that there are now two variable drug nodes, d_1 and d_2, and there is a negative edge between them as we want to avoid drug interaction between d_1 and d_2.

In this work, to allow flexibility in query expression, we assume the signs are given in the query graph. However, for specific applications, the signs may be inferred based on the types of the nodes adjacent to the edges. For example, for drug prescription, an edge between a drug node and an indication node should always be positive.

C. Answering Drug Queries

Given a drug graph and a query graph, answering query is to find a subgraph in the drug graph that matches the query graph. We formally define an answer to a query as follows. An *answer* of a query $Q(V_Q, E_Q, type_Q, key_Q)$ is in the form of a mapping function f that maps each query node to a node in the drug graph such that: (1) For each variable node q_v, $type_G(f(q_v)) = type_Q(q_v)$. (2) For each reference node q_r, $type_G(f(q_r)) = type_Q(q_r)$ and $key_G(f(q_r)) \cap key_Q(q_r) \neq \emptyset$.

Traditional query systems find exact answers for a given query, which are defined as follows. An *exact answer* is an answer f that satisfies the following properties: (1) For each positive edge $e(q_i, q_j)$ in E_Q, there is an edge $e(f(q_i), f(q_j))$ in the drug graph. (2) For each negative edge $e(q_i, q_j)$ in E_Q, there is no edge between $f(q_i)$ and $f(q_j)$ in the drug graph. In this work, to cope with data incompleteness, we consider both the answers that exactly match the query graph and those that *closely* match the query graph. Therefore, the main problems we address are as follows: (1) How to assign a score to an answer to quantify how well it matches the query? (2) How to find the top-k answers with the highest scores to present to the users? We present our solutions to these problems in the next section.

IV. Methodology

In this section, we describe how we design a score function to evaluate the quality of the answers and present our algorithm for finding the top-k answers according to the score function. To simplify the explanation, the solution presented in this section does not take into account patient profiles. We discuss how to extend the approach to provide personalization in Section V.

A. Design of the Score Function

Because the drug information graph is incomplete, having no edge between a drug node and a drug property node does not necessarily mean that there is no association between the node pair. Taking this fact into consideration, our score function quantifies for each node pair the likelihood that there is an edge between the node pair and integrates the likelihood into the score of an answer. We first describe how to quantify the likelihood of an edge and then present the score function.

1) Quantifying Edge Likelihood: Several approaches for predicting drug properties have been proposed, which may be applied to quantify the edge likelihood between a drug and a drug property. Most of the approaches apply supervised machine learning techniques, using different feature sets to represent drugs. In these approaches, for each targeted drug property p, a classifier that determines whether a drug has the property p is trained by using the drugs known to have property p as positive examples. A state-of-the-art approach proposed by Liu et al. uses chemical, biological, and phenotypic properties of drugs as drug features to predict drug side effects [30]. In this approach, each dimension of a drug feature vector is associated with a drug property p and has a binary value indicating whether the drug has the drug property p. We refer to this approach as the *fine-grained-feature* approach or the FG approach.

While the FG approach has shown great potential in predicting drug properties, our experiments (Section VII) show that its performance degrades when the number of positive examples of the targeted drug property is small. This is a common problem when the feature vector is high-dimensional. To improve the predicting performance for drug properties with small positive samples, we propose a network-based approach, which uses summarized features instead of fine-grained features to represent drugs. We describe our network-based approach in the following.

Network-based Approach. The network-based approach utilizes the structure of the drug information graph to create drug features. Intuitively, in the drug graph, two nodes that have many paths in-between should be closely related and more likely to have an edge between them. Thus, given a property node p, we can use the number of paths between the property node p and a drug node to quantify the likelihood that the drug will have property p. However, simply using the number of paths may not be effective. Because the drug graph contains several types of nodes, the paths in the graph are representing different types of complex relationships. These complex relationships can have different importance in indicating whether there is an edge between a given node pair. Therefore, we differentiate the paths by *path types* so that different importance levels can be assigned to different path types. A type of a path is defined from the types of nodes along the path. For example, the path d_1(D)-t_1(T)-d_2(D)-se_1(SE), where d_1, t_1, d_2, and se_1 are nodes in the drug graph, has the path type of D-T-D-SE, which represents the relationship where a drug shares a target with another drug that has a particular side effect.

Our approach uses the closeness between a drug and a targeted drug property according to a path type m as a feature of the drug. To quantify the closeness between a drug d and a drug property node p with respect to a path type m, we consider two metrics:

1) **Path count.** In this metric, the number of type-m paths between d and p is used as their closeness.
2) **Random walk.** This metric is based on a random walk. The closeness of node d to node p is the probability of being at node d when we perform a random walk starting from node p, going along the paths of type m in reverse direction. Thus, for path type m with length l, we consider an l-step random walk. In each step, only a node of the type specified in m are considered. For example, suppose path type m is D-T-D-SE. Then, starting from a node p (type SE), in the first step, the random walk moves to a type-D neighbor of p with the probability of $1/N_D(p)$, where $N_D(p)$ is the number of type-D neighbors of p. Similarly, in the next step, only the type-T neighbors of the current node, d', are considered for the visit with the probability of $1/N_T(d')$.

We model the likelihood of an edge between a drug d and a property node p as a function of the closeness between d and p with respect to multiple path types. Formally, let $M = \{m_1, ..., m_k\}$ be the set of path types. Let $c_{m_l}(v_i, v_j)$ be the closeness between d and p with respect to path type m_l. The likelihood of an edge between v_i and v_j, denoted by $p(v_i, v_j)$, is based on a logistic regression model as follows.

$$p(v_i, v_j) = 1/(1 + e^{-(\beta_0 + \beta_1 c_{m_1}(v_i,v_j) + ... + \beta_l c_{m_l}(v_i,v_j))}), \quad (1)$$

where $\beta_0, ..., \beta_l$ are the model parameters reflecting the importance of each path type.

Learning Model Parameters. For different types of edges (e.g., drug-side effect edge or drug-drug edge), the importance of each path type can be different. Furthermore, even for the same edge type, the importance of path types could be different for each individual drug property. Therefore, we consider two schemes for parameter learning.

1) **Per-type parameter learning**: Learn one set of parameters for each edge type.
2) **Per-node parameter learning**: Learn one set of parameters for each property node.

For example, with per-type learning, a single set of parameters is learned for the edge type drug-side effect; with per-node learning, one set of parameters is learned *for each side effect*. While per-node learning may provide the parameters that better fit each node, its performance may be limited when the targeted drug property has only a few positive samples. In contrast, in per-type learning, all the positive samples for the targeted edge type are combined and used in a single learning process. However, if the importance of the paths varies greatly among the nodes, per-type learning may not yield good performance. We compare the performance of the two schemes in our experiments.

Selecting Path Types. The path types used as features should correspond to the relationships that hint existence of the targeted edge type. In this work, our targeted edge types

include drug-side effect, drug-indication, and drug-drug. The path types used for the drug-side effect edges are shown in the second column of Table I. The path type D-D-SE (P1) is based on the hypothesis that drugs that interact tend to have similar side effects. The other path types are selected based on the hypothesis that if two drugs share a *property*, such as an indication or a target, they tend to have similar side effects; therefore, these paths are in the form of D-$propType$-D-SE, where $propType$ is a property node type, including I, P, T, SE, and CH. The path types used for drug-indication edges and drug-drug edges are selected with the same idea, as shown in Table I.

TABLE I
PATH TYPES USED FOR COMPUTING EDGE LIKELIHOOD.

Path ID	D-SE edge	D-I edge	D-D edge
P1	D-D-SE	D-D-I	D-D-D
P2	D-I-D-SE	D-I-D-I	D-I-D-D
P3	D-SE-D-SE	D-SE-D-I	D-SE-D-D
P4	D-P-D-SE	D-P-D-I	D-P-D-D
P5	D-T-D-SE	D-T-D-I	D-T-D-D
P6	D-CH-D-SE	D-CH-D-I	D-CH-D-D

Hybrid Approach. Since our network-based approach uses summarized features, which is less informative than the fine-grained features, when a targeted drug property has large positive training data, the FG approach could perform better. Therefore, we propose a hybrid approach that combines the FG approach and the network-based approach. In the hybrid approach, if a targeted drug property has more than τ known associated drugs, the FG approach is applied. Otherwise, the network-based approach is applied. We select τ to be 20 based on our experiments in Section VII.

2) Score Function for Query Answers: Based on our approach for quantifying the edge likelihood, we now define the score of an answer for a given query. Intuitively, in a good answer, the edges among the matches of the query nodes should correspond to the edges in the query graph. More specifically, if there is a positive edge between q_i and q_j, then there should be an edge between $f(q_i)$ and $f(q_j)$, the matches of q_i and q_j in an answer f. If there is a negative edge between q_i and q_j, then there should be no edges between $f(q_i)$ and $f(q_j)$. Our scoring function is defined based on this concept, as follows. For a given query graph Q, let E_Q^+ and E_Q^- denote the set of positive edges and the set of negative edges in the query graph, respectively. Let $w_G(v_i, v_j)$ be the function indicating whether there is an edge between v_i and v_j in the drug graph. That is, $w_G(v_i, v_j) = 1$ if there is an edge between v_i and v_j in the graph G. Otherwise, $w_G(v_i, v_j) = 0$. The score of an answer f, denoted by $S(f)$, is defined as

$$S(f) = \prod_{e(q_i,q_j)\in E_Q^+} p'(f(q_i), f(q_j)) \prod_{e(q_i,q_j)\in E_Q^-} (1 - p'(f(q_i), f(q_j))), \quad (2)$$

where $p'(v_i, v_j) = 1$ if $w_G(v_i, v_j) = 1$; otherwise, $p'(v_i, v_j) = p(v_i, v_j)$ (defined in Eq. 1). The function p' is used to adjust the edge likelihood score to 1 if an edge already exists in the graph. The first product in $S(f)$ considers the edge likelihood between the node pairs connected by positive edges. The second product considers the complement of the edge likelihood, i.e., $1 - p(v_i, v_j)$, for the node pairs connected by negative edges. The value of $S(f)$ ranges from 0 to 1. An answer f having $S(f)$ equal to 1 is an exact answer.

B. Finding the Top-k Answers

Having defined the score function, now we describe the algorithm for finding the top-k answers that have the highest scores. The algorithm consists of three main steps as follows.

Step 1: Find candidates matches of each query node. Based on the definition of an answer in Section III-C, the candidates of a reference node, q_r, are the nodes that have the same type as q_r and have at least one keyword that matches with $key_Q(q_r)$. The candidates of a variable node, q_v, are the nodes that have the same type as q_v. We denote the set containing all the candidate matches of a query node q_i as $can(q_i)$.

Step 2: Compute edge likelihood among candidate matches. The edge likelihood scores are used for computing the scores of the answers. For each edge in the query graph, $e(q_i, q_j)$, we compute the edge likelihood between the nodes in $can(q_i)$ and $can(q_j)$, which requires counting paths of different types among the candidate nodes. To count the paths of a specific type T, we use a modified breadth-first search (BFS) algorithm, where in each level of the search, only the nodes having the correct type according to T are visited. With a BFS starting from a source node v, we can obtain the number of T-paths from v to every node in the graph. Since we need one BFS for each path type, for each node in $can(q_i)$, we perform m BFSes, where m is the number of path types being used. In total, for an edge $e(q_i, q_j)$, we perform at most $m \cdot |can(q_i)|$ BFSes.

Step 3: Search for top-k answers. In this step, we search for k answers that have the highest scores among all the answers, which are all the combinations of the query nodes' candidate matches. We apply the branch-and-bound technique to obtain the top answers quickly. As the technique is a classic solution for combinatorial optimization, we refer readers to [31] for more details.

V. PERSONALIZING ANSWERS BASED ON PATIENT PROFILES

It is not uncommon that some drug properties are more common or present only in a patient with a specific profile. For example, the side effect *vomiting* for the drug *Tamiflu* is more common in children than adults [32]. In this section, we describe how to extend the approach in Section IV to use such information to personalize query answers.

A. Data Sources for Personalization

We assume that we have information on drug properties that are conditional on patient characteristics. While not much of such information are available yet, there are several data sources that can potentially be used to obtain such information. For example, the FDA adverse event reports contain information on side effects of drugs experienced by a patient along

with his/her information including age, gender, and weight. Another potential data sources are online communities, such as Facebook, Twitter, and health-related online message boards. Existing studies have shown promising results in using these resources to learn about side effects of drugs [33]–[41]. These online sources usually contain basic user profiles, such as age, gender, and location. Additionally, based on what people have expressed online, habits and lifestyles, such as being a vegetarian, smoking, or regular exercising, may be inferred. These data sources could allow future research to extract correlation between patient characteristics and drug properties.

From such data sources, we assume that we have a set of tuples $\mathcal{T} = \{(v_i, v_j, A_k, val)\}$, where v_i is a drug node, v_j is a drug property node, A_k is a patient attribute, and val is a value of A_k. Each tuple (v_i, v_j, A_k, val) indicates that a drug v_i exhibits a property v_j for a patient whose attribute A_k is equal to val.

B. Patient Profiles

A *patient profile* is used to describe a patient. Let $A_1, ..., A_n$ be all the attributes presented in $\mathcal{T}$. We represent a patient profile as a vector $\vec{A}$ of size n, where $\vec{A}[i]$ is a value corresponding to attribute A_i of the patient. We assume that the domain of each attribute is categorical. Attributes that are originally numerical can be transformed to categorical by grouping into ranges. For example, age can be divided into four ranges: child (age 0-12), teenage (age 13-19), adult (age 20-64), and elder (age 65 and up).

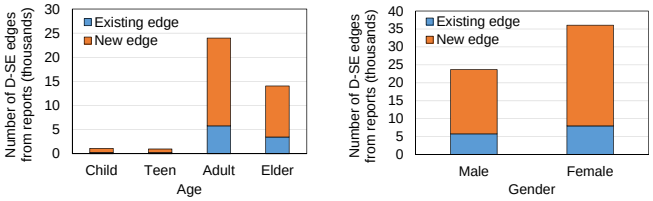
C. Personalizing Answers

To provide personalized answers for a query with patient profile $\vec{A}$, we create a *personalized drug graph*, $G_{\vec{A}}$, that contains information about drug properties with respect to the profile $\vec{A}$. Let G represent a *core drug graph*, containing relationships between drugs and drug properties that do not rely on patient characteristics. $G_{\vec{A}}$ is constructed by using data from $\mathcal{T}$ to modify the edges in G. Initially, we let $G_{\vec{A}}$ contains all the nodes and edges in G. Then, for each tuple (v_i, v_j, A_k, val) in $\mathcal{T}$, if $\vec{A}[k] = val$, an edge (v_i, v_j) is added to $G_{\vec{A}}$.

With $G_{\vec{A}}$, the approach described in Section IV can be applied to obtain personalized answers. Since $G_{\vec{A}}$ contains the drug properties with consideration of patient profiles, the answer obtained is tailored for the specific patient with profile $\vec{A}$.

D. Case study: Personalization on Side effects of Drugs

As a case study, we describe how we personalize the answers on side effects of drugs. We use FDA adverse event reports as a side effect personalization data source. Each report contains the information about a patient (age, gender, and weight), the list of drugs taken by the patient, and the side effects of the drugs. The reports are submitted by both healthcare professionals and consumers. Although we cannot conclude that the drugs in each report are the causes of the side effects, the reports provide signals of potential associations between side effects and drugs for different patient profiles.



(a) For age attribute (b) For gender attribute

Fig. 2. Number of drug-side effect edges obtained from FDA reports.

Our core drug graph contains only the associations between drugs and side effects that are common for all patients, for example, those extracted from drug labels. Using the reports, we add a tuple (d, se, A, val) to $\mathcal{T}$ if there are at least T reports that indicate a patient with attribute $A = val$ has experienced the side effect se when using the drug d, where T is a threshold value. We focus on two patient attributes, age and gender. As mentioned earlier, age is grouped into four ranges. In Figure 2, we show the number of drug-side effect edges obtained from the reports with respect to each attribute and its value. The figure shows both the number of edges that are already in the core graph and those that are not. It can be seen that using the FDA reports, we can obtain a large number of additional drug-side effect associations that may be linked to a particular patient attribute.

E. Case study: Personalization with Biomarker Data

Genetic traits in patients can effect drug effectiveness and side effects. Some drugs are designed for treating diseases with specific biomarkers. To include this information in our query system, we use the table of drug pharmacogenomic biomarkers provided by FDA[1]. By manually going through the drug labels, we obtain 98 personalized tuples from this data source and add them to $\mathcal{T}$. These tuples include drug indications that are specific to a particular biomarker and side effects that are introduced if patients have specific biomarkers. From the drug labels, we can also obtain information on the need of dosage adjustment in case some biomarkers are presented. While currently our system does not provide dosage recommendations, these data can potentially be utilized to provide personalized dosage recommendations.

VI. Drug Query System Prototype

In this section, we describe the drug query system prototype that we have developed.

A. System Overview

The overview of the drug query system is shown in Figure 3. The system has four main components: drug information base, query processor, query translator, and user interface. The user interface accepts a drug query from a user and displays the answers obtained from the query processor to the user. The query translator is responsible for translating an input query, which is in a user-friendly format, to its corresponding query graph. The query processor computes the answers for the query graph using the algorithm we have described.

[1]http://www.fda.gov/drugs/scienceresearch/researchareas/pharmacogenetics/ucm083378.htm

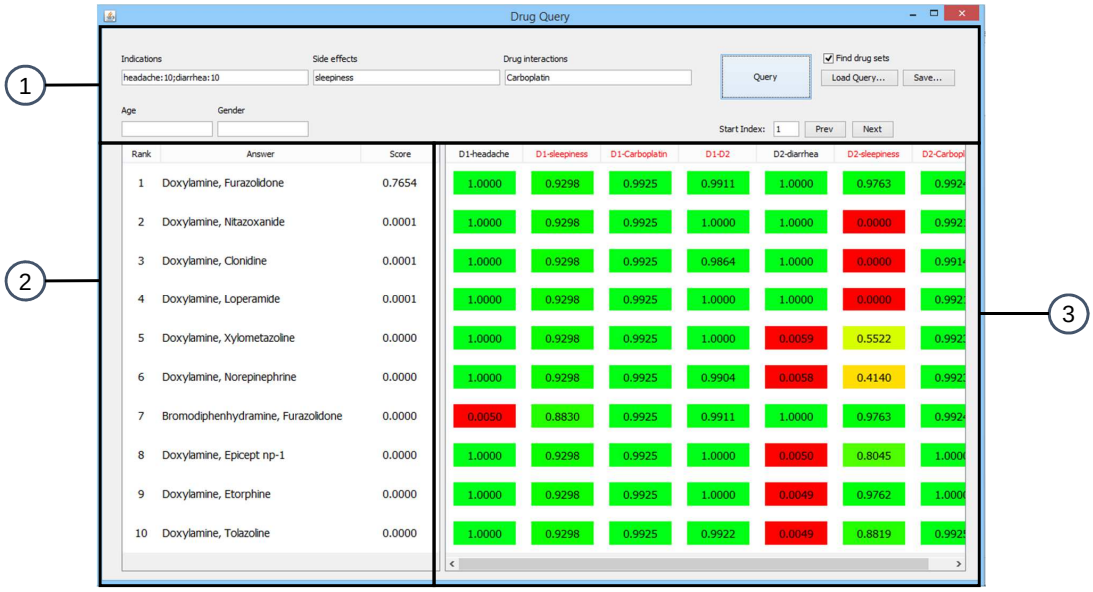


Fig. 4. User Interface of Drug Query System

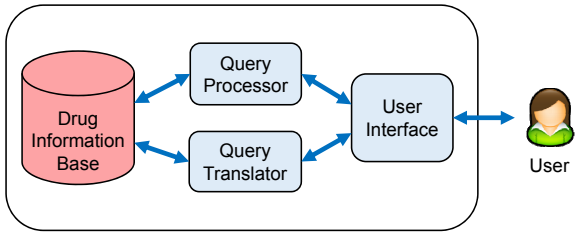


Fig. 3. Drug query system

B. User Interface

The user interface of the system is shown in Figure 4. The user interface provides a form for the user to input the conditions for the drug query in terms of drug indications, side effects, and drug interactions, along with a patient profile (No. 1 in Figure 4). For each type of conditions, the user can input keywords, each of which will be mapped to nodes in the drug information graph by the query translator. The answers of a query and their scores are shown in a result table (No. 2 in Figure 4).

In a drug query, some conditions may be more crucial to the users than the others, while the answers are ranked based on an aggregated score that captures how well the answers can satisfy all the conditions. Considering the subjective nature of the drug queries, to help users find the answers that best suit their needs, we provide two additional features: score component visualizer and condition weight adjustment.

Score Component Visualizer. A score component visualizer (No. 3 in Figure 4) displays how well an answer can satisfy each of the conditions in the query. The condition scores are color-coded to allow users to quickly see the overview the results. For example, in Figure 4, the user queries for a drug for headache and a drug for diarrhea without the sleepiness side effect and without drug interaction with Carboplatin. The sixth column in the score component visualizer corresponds to whether the second drug in an answer has sleepiness slide effect. By scanning the column, the user can quickly see that the second and third answers are likely to cause sleepiness. If sleepiness needs to be strictly avoided, the user can decide to leave the drugs out from their options.

Condition Weight Adjustment. Condition weight adjustment allows user to increase the importance of some conditions in the query. By default, every condition has a weight of 1, which means they are treated equally. By assigning higher weights to some conditions, the query processor will adjust the ranking of the answers to bias towards those that can better satisfy the conditions with high weights. The weight adjustment can be done by tagging a weight with the keyword of a condition. For example, to assign a weight of 10 to the headache indication, the user can put ”headache:10” into the indication input field.

VII. Evaluation

Our evaluation consists of two parts. First, we evaluate the quality of edge likelihood scores. Then, we evaluate our query system by showing examples of query results and discuss the benefits provided by our system.

A. Data Sources and Drug Graph Characteristics

We consolidate drug information from multiple data sources to create the drug graph for our prototype query system. The details of each data source are given as follows.

DrugBank. DrugBank [2] is a database that contains chemical, pharmacological, and pharmaceutical data of drugs along with drug target information. For each drug, we obtain its targets, drug interactions, and chemical substructure signature. The number of drugs in DrugBank is 7,682. 86% of the drugs have target information. 15% of the drugs have the drug interaction information.

SIDER2. SIDER2 [3] contains the information about side effects extracted from drug labels. The side effects are mapped to

MedDRA[2] preferred terms. There are 2,021 drugs in SIDER2. 49% of the drugs have side effect information.

KEGG Drug. KEGG Drug [4] is a database containing information for approved drugs in Japan, USA, and Europe. For each drug, we obtain the information of its associated pathways. There are 9,354 drugs in the database. 27% of the drug have pathway information.

NDF-RT. The National Drug File-Reference Terminology (NDF-RT) is a part of the Unified Medical Langage System (UMLS) [42], which defines and provides connections between medical terms. We use the relationships between drugs and diseases (indications) extracted from NDF-RT by Wang et al. [43]. The dataset contains 799 drugs and 719 diseases.

From the above data sources, we link the information of each drug by matching drug brand names and generic names in each of the data sources and create the core drug graph. We remove the drug nodes that do not have any links to the other nodes. The resulting graph contains 16,565 nodes and 256,447 edges. The numbers of nodes and edges of each type are shown in Table II. To create a personalized graph, we use the adverse drug event reports and the biomarker data from drug labels, provided by FDA, as described in Section V.

TABLE II
DRUG GRAPH CHARACTERISTICS.

Node Type	#Nodes
Drug	7,705
Indication	711
Side effect	3,034
Pathway	131
Target	4,103
Chem. Sub.	881
Total	16,565

Edge Type	#Edges
Drug-Drug	24,224
Drug-Indication	3,210
Drug-Side Effect	93,158
Drug-Pathway	2,427
Drug-Target	15,105
Drug-Chem. Sub.	118,323
Total	256,447

B. Evaluation of Edge Likelihood Quantification

Edge likelihood scores are the basis for computing the scores of the answers. The accuracy of the likelihood scores directly affects the quality of the query answers. Therefore, it is important that the likelihood scores are good predictors of the existence of edges.

1) Evaluation Method: We evaluate the edge likelihood scores obtained from six approaches:

1) Fine-grained-feature approach (FG)
2) Network-based approach with path count metric and per-node parameter learning (MP)
3) Network-based approach with path count metric and per-type parameter learning (MPG)
4) Network-based approach with random walk metric and per-node parameter learning (MP_RW)
5) Network-based approach with random walk metric and per-type parameter learning (MPG_RW)
6) Hybrid approach (Hybrid): combination of FG and MPG_RW

For all of these approaches, we build logistic regression models by utilizing the Python package scikit-learn[3]. The evaluation is performed for three types of edges: drug-side effect (D-SE), drug-indication (D-I), and drug-drug (D-D). For each edge type t, we perform 10-fold cross-validation. In each fold, 10% of the drugs are assigned as test drugs, used for evaluating the performance of the predictors. We use AUROC and AUCPR as the evaluation metrics. AUROC is the area under the ROC curve, the plot of the true positive rate against the false positive rate computed from different threshold values of the likelihood score. AUCPR is the area under the precision-recall curve. Both metrics are commonly used in machine learning. While AUROC emphasizes on identifying both the positives and negatives correctly, AUCPR focuses more on the performance of an approach in identifying true positives [44]. We therefore use both metrics in our evaluation. The value of AUROC and AUCPR is between 0 and 1. The higher the value, the better.

TABLE III
PREDICTING PERFORMANCE COMPARISON

	Algorithm					
	FG	**MP**	**MPG**	**MP_RW**	**MPG_RW**	**Hybrid**
AUROC						
D-SE	0.7409	0.7503	0.8400	0.8276	0.8658	**0.8753**
D-I	0.8311	0.7938	0.8835	0.9166	0.9166	**0.9171**
D-D	0.7211	0.7546	0.7871	0.8219	0.8289	**0.8290**
AUCPR						
D-SE	0.3807	0.3285	0.3609	0.3177	0.3985	**0.4302**
D-I	0.3857	0.3370	0.4668	0.5527	0.5679	**0.5723**
D-D	0.3463	0.3500	0.3153	0.3863	0.3945	**0.3978**

2) Performance Comparison: Table III shows the performance of the six edge likelihood prediction approaches. First, we compare FG, MP, and MP_RW. It can be seen that the MP approach has competitive performance with the FG approach. Each of them performs better than the other in some of the edge types. On the other hand, the MP_RW approach significantly outperforms both the FG and MP approaches in terms of AUROC. It also performs better in terms of AUCPR for the D-I and D-D edges. These results show that the network-based features can potentially improve prediction performance and that the closeness metrics used in the network-based approaches can greatly affect the predicting performance.

Next, we compare the two schemes of parameter learning in the network-based approaches. The results show that the approaches with per-type parameter learning tend to perform better than their per-node-learning counterparts (MP vs. MPG and MP_RW vs. MPG_RW). This illustrates the advantage of per-type parameter learning, where the positive samples from all the drug properties can be combined. Finally, it can be seen that the Hybrid approach has the best performance among all the six approaches. Comparing to the FG approach, the improvements in AUROC are 18%, 10%, and 15% for edge type D-SE, D-I and D-D, respectively, and the improvements in AUCPR are 13%, 43%, and 15% for edge type D-SE, D-I and D-D, respectively.

The above results may be counterintuitive in that the network-based approaches have better performance than the FG approach despite using summarized features of drugs. To better understand why this is the case, we investigate

[2]http://www.meddra.org
[3]http://scikit-learn.org/

the performance of the edge likelihood prediction approaches with varying numbers of positive training samples. For each property type (SE, I, D), we group the properties based on their numbers of positive training samples and report the prediction performance for each group. The result is shown in Figure 5.

From the figure, we observe that the network-based approaches outperform the FG approach when the number of training samples is small. The improvement is especially significant for drug properties with less than 20 positive training samples, where we can see the improvement in both AUROC and AUCPR. For such drug properties, compared to the FG approach, the MPG_RW approach improves the AUROC by 37%, 11%, and 21% and the AUCPR by 46%, 30%, and 43% for the edge type D-SE, D-I and D-D, respectively. As the number of positive training samples becomes larger, the performance of the FG increases and finally overcomes the network-based approaches. The points at which the FG approach starts to perform better vary among the network-based approaches and different edge types. For the MPG_RW approach, the FG approach requires 60 positive examples for each drug property to obtain better AUROC and 20 positive examples to obtain better AUCPR. Therefore, in the Hybrid approach, we set the threshold for switching prediction algorithms to be 20 positive samples.

In Figure 6, we show the percentage of drug properties in each group of training sample sizes. It can be seen that the majority of drug properties have less than 20 positive training samples. This explains why the overall performance of the network-based approaches is better than the FG approach (as in Table III). Additionally, these results illustrate the benefit of the Hybrid approach, which selects the appropriate approaches to use based on the size of the positive training data.

3) Benefits from Using Multiple Path Types: We evaluate how much using multiple path types helps to improve the performance, compared to using a single path type. Figure 7 shows the performance when each of the path type is used individually and when all the six path types (listed in Table I) are used. The results show that using multiple types of paths can significantly improve the performance. When a single path type is used, the maximum AUROC that can be achieved are 0.8651 (P3), 0.8743 (P2), and 0.7804 (P5), for D-SE, D-I, and D-D edges, respectively. When all the six path types are used, we can obtain the AUROC of 0.8658, 0.9166, and 0.8289. Additionally, it should be noted that the difference in the AUROC obtained from each path type supports our hypothesis that different path types have unequal importance in signifying the likelihood of edge existence.

C. Evaluation of Query Answering

In this section, we demonstrate the usefulness of our query system by showing examples of the query results returned from our system.

First, we show the top results for the query **[Query 1]** *"Find a drug for tonsillitis*[4]*"* in Table IV. Our system finds both exact matches and close matches for this query. The top four answers, which include Cefdinir, Ceftibuten, Azithromycin,

[4]an inflammation of tonsils caused by bacteria or virus infection

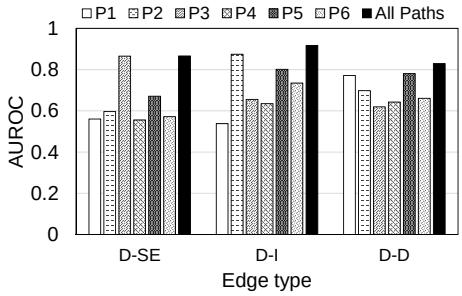


Fig. 7. Performance comparison when a single path type is used and when all the path types are used.

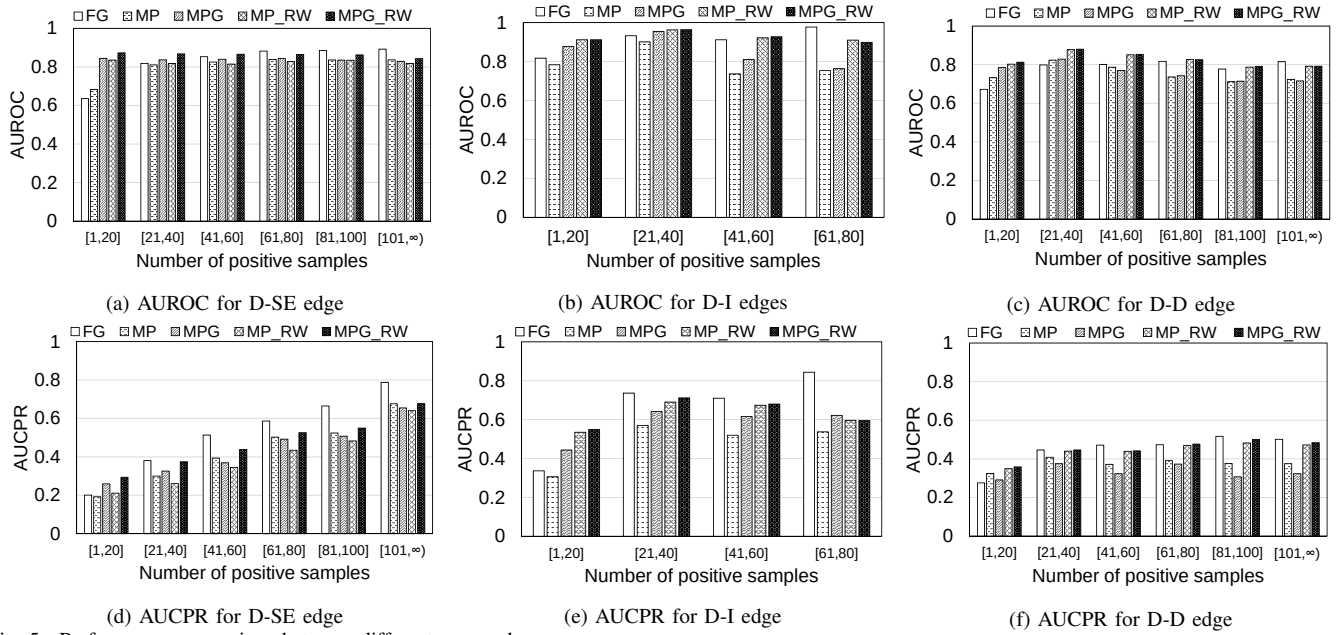
TABLE IV
RESULTS FOR QUERY 1.

Rank	Drug	Score	Answer quality
1	Cefdinir	1	Relevant
2	Ceftibuten	1	Relevant
3	Azithromycin	1	Relevant
4	Cefpodoxime	1	Relevant
5	Cefaclor	0.11	Relevant
6	Cefixime	0.11	Relevant
7	Moxifloxacin	0.08	Relevant
8	Cefprozil	0.07	Relevant
9	Levofloxacin	0.07	Relevant
10	Ceftazidime	0.07	No direct support

and Cefpodoxime, have the highest possible score of 1. This means according to the drug information graph, these drugs are indicated for tonsillitis and thus they exactly match the query. The answers below the fourth place are inexact matches. According to the drug graph, these answers are not indicated for tonsillitis. However, by manually checking with external online data sources, we found supporting evidence that Cefaclor, Cefixime, Moxifloxacin, Cefprozil, and Levofloxacin, which are ranked from the fifth to the ninth places, may be used to treat tonsillitis [45]–[48]. For the tenth drug, Ceftazidime, although we cannot find references for its use in treating tonsillitis, it is also an antibacterial drug according to RxList[5]. This example illustrates that our approach can provide answers that exactly match the query as well as inexact matches that are potentially useful for users, which provides users with more alternatives. Nevertheless, it should be noted that the query system is not intended to replace experts but to assist them in finding drugs that suit their needs.

Next, we consider the following query: **[Query 2]** *"Find a drug for schizophrenia for the patient who is taking Paroxetine."* In Table V, we compare the results from the traditional approach that finds only exact matches and the top results from our approach. Using the traditional approach, we obtain exactly 13 drugs. There are no ranking among these drugs since only the exact matches are returned. We manually checked drug interactions on external data sources, Drugs.com[6] and Medscape[7], and found that in fact all drugs except Reserpine and Deserpidine can interact with Paroxetine. However, such interaction data are not contained in our drug interaction data source (DrugBank) and therefore the drugs

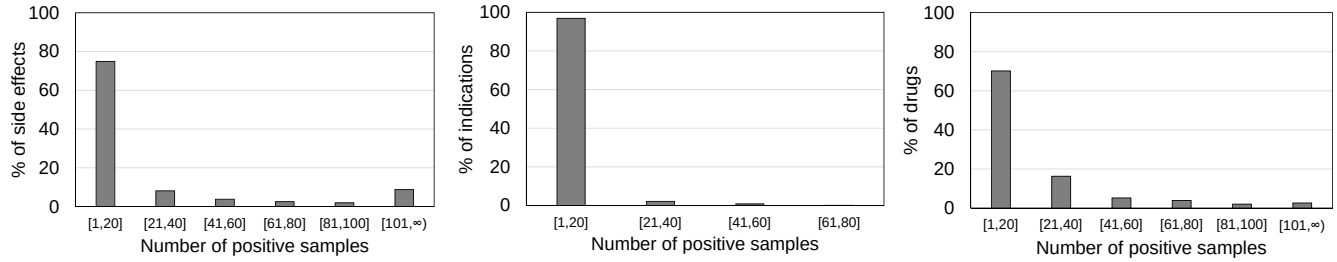
[5]http://www.rxlist.com
[6]http://www.drugs.com
[7]http://www.medscape.com/



(a) AUROC for D-SE edge (b) AUROC for D-I edges (c) AUROC for D-D edge

(d) AUCPR for D-SE edge (e) AUCPR for D-I edge (f) AUCPR for D-D edge

Fig. 5. Performance comparison between different approaches.



(a) D-SE (b) D-I (c) D-D

Fig. 6. Percentage of drug properties in each group of positive training sample sizes

are returned as answers. Using our approach, the returned answers receive varying scores, and the scores are less than one, indicating the possibility that they might interact with Paroxetine. Additionally, the two drugs that do not interact with Paroxetine according to our external data sources are ranked at the first place and the third place in our result list. This example demonstrates that by taking into account the likelihood of edges, our system is more informative and can help users to discover the drugs that fit their requirements better.

In the next two examples, we illustrate how the answers are personalized according to a given patient profile. We use the query: **[Query 3]** *"Find a drug for schizophrenia without the side effect of cardiac arrest."* We compare the top 15 results obtained when a user profile is not given and when the user profile is specified as {*female, elder*} in Table VI. When the user profile is given, three of the drugs, which are Methotrimeprazine, Haloperidol, and Asenapine, are removed from the list. These three drugs were reported (via the FDA drug adverse event reporting system) as potential causes of cardiac arrest in elder female patients, and our approach takes into account this fact and adjusts the scores of the drugs accordingly. Another example is **[Query 4]** *"Find a drug for malaria falciparum without the side effect of hemolysis."* We compare the top 5 results obtained for a normal patient and a patient with G6PD deficiency[8] in Table VII. It can be seen that the drug Primaquine is replaced by Pyrimethamine when a patient has G6PD deficiency. This is because it has high risk of causing hemolysis in such patients.

Finally, we show an example of a query that asks for

TABLE V
RESULTS FOR QUERY 2.

(a) EXACT MATCH ONLY

Drug
Trifluoperazine
Carbamazepine
Perphenazine
Fluphenazine
Ziprasidone
Olanzapine
Haloperidol
Reserpine
Clozapine
Molindone
Cyproheptadine
Aripiprazole
Deserpidine

(b) OUR APPROACH

Rank	Drug	Score
1	**Deserpidine**	0.98
2	Cyproheptadine	0.96
3	**Reserpine**	0.95
4	Molindone	0.94
5	Carbamazepine	0.93
6	Fluphenazine	0.86
7	Trifluoperazine	0.84
8	Paliperidone	0.79
9	Haloperidol	0.78
10	Perphenazine	0.77
11	Ropinirole	0.76
12	Promazine	0.69
13	Aripiprazole	0.65

[8]a genetic disorder characterized by abnormally low levels of the enzyme G6PD

TABLE VI
RESULTS FOR QUERY 3.

(a) WITHOUT PROFILE

Rank	Drug	Score
1	Deserpidine	0.97
2	Mesoridazine	0.95
3	Reserpine	0.93
4	Molindone	0.92
5	**Methotrimeprazine**	0.87
6	Cyproheptadine	0.87
7	Chlorpromazine	0.84
8	**Haloperidol**	0.80
9	Promazine	0.71
10	**Asenapine**	0.68

(b) FOR AN ELDER FEMALE.

Rank	Drug	Score
1	Deserpidine	0.97
2	Mesoridazine	0.95
3	Reserpine	0.93
4	Molindone	0.92
5	Cyproheptadine	0.87
6	Chlorpromazine	0.84
7	Promazine	0.71
8	Cabergoline	0.53
9	Loxapine	0.53
10	Apomorphine	0.53

TABLE VII
RESULTS FOR QUERY 4.

(a) WITHOUT G6PD DEFICIENCY.

Rank	Drug	Score
1	Proguanil	0.99
2	Sulfadoxine	0.99
3	Quinine	0.99
4	Quinacrine	0.99
5	**Primaquine**	0.99

(b) WITH G6PD DEFICIENCY.

Rank	Drug	Score
1	Proguanil	0.99
2	Sulfadoxine	0.99
3	Quinine	0.99
4	Quinacrine	0.99
5	**Pyrimethamine**	0.99

multiple drugs as follows: **[Query 5]** *"Find a set of drugs for Parkinson's disease (d_1), myositis (d_2), and depression (d_3), that do not interact with one another."* If a traditional exact match approach is used, more than one thousand drug sets would be returned for this query, which can make it difficult for users to select the best answer. In Table VIII, we show the top 10 results obtained from our approach. Using our approach, some of the drug sets receive lower scores because of their potential interactions. Upon verifying the drug interactions manually on Drugs.com, we found that there are no evidence of interactions among drugs in the top four drug sets. For the drug sets in the lower ranks, each of them contains a drug pair that may interact with each other, as shown in Table VIII.

VIII. DISCUSSION

This work addresses an important problem in supporting drug queries, which is how to obtain and rank answers based on incomplete information and provide personalization. Here we discuss several directions for future work that lead towards a complete and practical drug query system.

We developed a prototype query system to demonstrate the effectiveness of our approach. The system answers drug queries based on the input drug information graph. As drug data are continually updated, e.g, new drugs are added or new side effects are discovered, the drug information graph should be periodically recreated from a new snapshot of data. In practice, it would be convenient to develop a process that automatically downloads raw data from data sources and creates an updated drug graph at specified time intervals.

TABLE VIII
RESULTS FOR QUERY 5.

Rank	Drug1 (d_1)	Drug2 (d_2)	Drug3 (d_3)	Answer Quality
1	Carbidopa	Chlorzoxazone	Isoflurane	No interactions
2	Carbidopa	Orphenadrine	Isoflurane	No interactions
3	Amantadine	Chlorzoxazone	Isoflurane	No interactions
4	Amantadine	Chlorzoxazone	Methylphenidate	No interactions
5	Ropinirole	Chlorzoxazone	Isoflurane	$d_1 \leftrightarrow d_2$
6	Pramipexole	Chlorzoxazone	Isoflurane	$d_1 \leftrightarrow d_3$
7	Ropinirole	Chlorzoxazone	Ropinirole	$d_1, d_3 \leftrightarrow d_2$
8	Carbidopa	Chlorzoxazone	Temazepam	$d_2 \leftrightarrow d_3$
9	Carbidopa	Chlorzoxazone	Lorazepam	$d_2 \leftrightarrow d_3$
10	Biperiden	Chlorzoxazone	Isoflurane	$d_1 \leftrightarrow d_2$

While the focus of this paper is on the quality of the answers, the response time of the drug query system is also important. In general, the response time increases with the number of drugs that are asked for and the number of drug properties specified in the queries. With the current implementation, drug queries that look for a single drug with up to 10 drug properties can be answered within a few seconds. For drug queries that looks for two or more drugs, the system takes from one minute to several minutes. To make the system more practical, the response time can be improved by adopting pruning algorithms and leveraging parallel computation such as applying algorithms proposed in [13]–[16].

In addition to side effects and drug interactions, there are several factors that are important in prescribing drugs, for example, contraindications, drug pregnancy categories, and drug allergy. If data is available, our approach could be extended to support these additional factors. Finer-grained information, such as the incident rates of the side effects provided on drug labels, may also be taken into account to provide a better ranking of the drugs. One way to incorporate such information is to represent drug information with a weighted graph; however, an approach to combine such weights with the predicted edge likelihood needs to be studied. Additionally, there are existing clinical decision support systems that utilize electronic health records to provide drug recommendation [11], [12]. It would be interesting to study how to integrate our approach with such systems in order to provide a better recommendation.

It is also important that the user interface for the drug query system is user-friendly and informative. With the advances in natural language processing techniques, a module that translates natural language queries to query graphs could be developed. The system could be extended to provide warnings about severe risks in using drugs, for example, warnings about life-threatening side effects for patients with specific biomarkers or pregnant patients. This would be helpful to avoid associated risks in using drugs. Furthermore, the query system could potentially be integrated with an electronic medical record system so that patient information and conditions on drugs can be automatically imported or created. Our current system allows assigning different levels of importance to the conditions in the queries. A possible future direction is to automatically assign appropriate weights to the conditions, which may be achieved with assistance from domain experts and/or applications of machine learning techniques.

As mentioned in our evaluation, our system is designed to work with humans in finding the answers that fit their needs. It could be helpful to extend the system to show supporting evidence, such as published research articles or news, for the drug properties derived from prediction, which could shorten the time needed to review the answers. Additionally, it is possible to present the predicted drug properties to experts

and accept the feedbacks, which can then be used to update the underlying data source and improve the prediction model.

IX. Conclusion

In this paper, we propose an approach for answering drug queries to support drug prescription. To cope with incomplete and noisy data, we allow both exact and close matches when answering queries. The answers are ranked by utilizing the structure of a drug information network to quantify the likelihood of associations between drug and drug properties in the case that the associations are missing or unknown. We also present an intuitive approach to display answers to users, which aims to help users to understand the ranked results and possibly refine their queries. We demonstrate how our approach could assist professionals and drug consumers to make informed decision through several examples.

Acknowledgment

This work is supported by the NSF under Grant No. CNS-1217284 and CCF-1018114.

References

[1] S. Khemmarat and L. Gao, "Supporting drug prescription via predictive and personalized query system," in *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare*. IEEE, 2015.

[2] C. Knox *et al.*, "Drugbank 3.0: a comprehensive resource for omics research on drugs," *Nucleic acids research*, vol. 39, no. suppl 1, pp. D1035–D1041, 2011.

[3] M. Kuhn *et al.*, "A side effect resource to capture phenotypic effects of drugs," *Molecular systems biology*, vol. 6, no. 1, p. 343, 2010.

[4] M. Kanehisa and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.

[5] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[6] K. Sangkuhl *et al.*, "Pharmgkb: understanding the effects of individual genetic variants," *Drug metabolism reviews*, vol. 40, no. 4, pp. 539–551, 2008.

[7] A. Langer, R. Banga, A. Mittal, L. V. Subramaniam, and P. Sondhi, "A text based drug query system for mobile phones," *Int. J. Mob. Commun.*, vol. 12, no. 4, pp. 411–429, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1504/IJMC.2014.063656

[8] C. Doulaverakis *et al.*, "Panacea, a semantic-enabled drug recommendations discovery framework," *J. Biomed. Semant.*, vol. 5, p. 13, 2014.

[9] M. Dumontier and N. Villanueva-Rosales, "Towards pharmacogenomics knowledge discovery with the semantic web," *Briefings in bioinformatics*, vol. 10, no. 2, pp. 153–163, 2009.

[10] A. Ben Abacha and P. Zweigenbaum, "Medical question answering: translating medical questions into sparql queries," in *Proceedings of the 2nd ACM SIGHIT*. ACM, 2012, pp. 41–50.

[11] R. Kaushal, K. G. Shojania, and D. W. Bates, "Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review," *Archives of internal medicine*, vol. 163, no. 12, pp. 1409–1416, 2003.

[12] E. Ammenwerth, P. Schnell-Inderst, C. Machan, and U. Siebert, "The effect of electronic prescribing on medication errors and adverse drug events: a systematic review," *Journal of the American Medical Informatics Association*, vol. 15, no. 5, pp. 585–600, 2008.

[13] J. Jin, S. Khemmarat, L. Gao, and J. Luo, "Querying web-scale information networks through bounding matching scores," in *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, 2015, pp. 527–537. [Online]. Available: http://doi.acm.org/10.1145/2736277.2741131

[14] ——, "A distributed approach for top-k star queries on massive information networks," in *20th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2014, Hsinchu, Taiwan, December 16-19, 2014*, 2014, pp. 9–16. [Online]. Available: http://dx.doi.org/10.1109/PADSW.2014.7097785

[15] A. Khan, N. Li, X. Yan, Z. Guan, S. Chakraborty, and S. Tao, "Neighborhood based fast graph search in large networks," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 2011, pp. 901–912.

[16] A. Khan, Y. Wu, C. C. Aggarwal, and X. Yan, "Nema: Fast graph search with label similarity," in *Proceedings of the VLDB Endowment*, vol. 6, no. 3. VLDB Endowment, 2013, pp. 181–192.

[17] J. Scheiber *et al.*, "Mapping adverse drug reactions in chemical space," *Journal of medicinal chemistry*, vol. 52, no. 9, pp. 3103–3107, 2009.

[18] A. Bender *et al.*, "Analysis of pharmacology data and the prediction of adverse drug reactions and off-target effects from chemical structure," *ChemMedChem*, vol. 2, no. 6, pp. 861–873, 2007.

[19] F. Hammann *et al.*, "Prediction of adverse drug reactions using decision tree modeling," *Clinical Pharmacology & Therapeutics*, vol. 88, no. 1, pp. 52–59, 2010.

[20] E. Pauwels, V. Stoven, and Y. Yamanishi, "Predicting drug side-effect profiles: a chemical fragment-based approach," *BMC bioinformatics*, vol. 12, no. 1, p. 169, 2011.

[21] M. Fukuzaki *et al.*, "Side effect prediction using cooperative pathways," in *BIBM'0*. IEEE, 2009, pp. 142–147.

[22] L.-C. Huang, X. Wu, and J. Y. Chen, "Predicting adverse side effects of drugs," *BMC genomics*, vol. 12, no. Suppl 5, p. S11, 2011.

[23] J. Sun, H. Xu, and Z. Zhao, "Network-assisted investigation of antipsychotic drugs and their targets," *Chemistry & biodiversity*, vol. 9, no. 5, pp. 900–910, 2012.

[24] F. Cheng *et al.*, "Prediction of polypharmacological profiles of drugs by the integration of chemical, side effect, and therapeutic space," *Journal of chemical information and modeling*, vol. 53, no. 4, pp. 753–762, 2013.

[25] J. Sun *et al.*, "Network-assisted prediction of potential drugs for addiction," *BioMed research international*, vol. 2014, 2014.

[26] ——, "Characterization of schizophrenia adverse drug interactions through a network approach and drug classification," *BioMed research international*, vol. 2013, 2013.

[27] J. Huang *et al.*, "Systematic prediction of pharmacodynamic drug-drug interactions through protein-protein-interaction network," *PLoS computational biology*, vol. 9, no. 3, p. e1002998, 2013.

[28] F. Cheng and Z. Zhao, "Machine learning-based prediction of drug–drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties," *J. Am. Med. Inform. Assoc.*, vol. 21, no. e2, pp. e278–e286, 2014.

[29] H. Zheng *et al.*, "Linking biochemical pathways and networks to adverse drug reactions," *NanoBioscience, IEEE Transactions on*, vol. 13, no. 2, pp. 131–137, 2014.

[30] M. Liu *et al.*, "Large-scale prediction of adverse drug reactions using and phenotypic properties of drugs," *J. Am. Med. Inform. Assoc.*, vol. 19, no. e1, pp. e28–e35, 2012.

[31] J. Clausen, "Branch and bound algorithms-principles and examples," *Dept of Comp. Sci., University of Copenhagen*, pp. 1–30, 1999.

[32] Genentech USA, "Tamiflu (oseltamivir phosphate) Prescribing Information." http://www.tamiflu.com/hcp/prescribing/hcp_prescribe.jsp.

[33] R. Leaman, L. Wojtulewicz, R. Sullivan, A. Skariah, J. Yang, and G. Gonzalez, "Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks," in *Proceedings of the 2010 workshop on biomedical natural language processing*. Association for Computational Linguistics, 2010, pp. 117–125.

[34] J. Bian, U. Topaloglu, and F. Yu, "Towards large-scale twitter mining for drug-related adverse events," in *Proceedings of the 2012 international workshop on Smart health and wellbeing*. ACM, 2012, pp. 25–32.

[35] X. Liu and H. Chen, "Azdrugminer: an information extraction system for mining patient-reported adverse drug events in online patient forums," in *Smart Health*. Springer, 2013, pp. 134–150.

[36] C. C. Freifeld, J. S. Brownstein, C. M. Menone, W. Bao, R. Filice, T. Kass-Hout, and N. Dasgupta, "Digital drug safety surveillance: monitoring pharmaceutical products in twitter," *Drug Safety*, vol. 37, no. 5, pp. 343–350, 2014.

[37] S. Tuarob, C. S. Tucker, M. Salathe, and N. Ram, "An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages," *Journal of biomedical informatics*, vol. 49, pp. 255–268, 2014.

[38] A. Sarker and G. Gonzalez, "Portable automatic text classification for adverse drug reaction detection via multi-corpus training," *Journal of biomedical informatics*, 2014.

[39] A. Patki, A. Sarker, P. Pimpalkhute, A. Nikfarjam, R. Ginn, K. OConnor, K. Smith, and G. Gonzalez, "Mining adverse drug reaction signals from social media: going beyond extraction," *Proceedings of BioLinkSig*, vol. 2014, 2014.

[40] R. Ginn, P. Pimpalkhute, A. Nikfarjam, A. Patki, K. OConnor, A. Sarker, and G. Gonzalez, "Mining twitter for adverse drug reaction mentions: a corpus and classification benchmark," in *Proceedings of the fourth workshop on building and evaluating resources for health and biomedical text processing*, 2014.
[41] A. Nikfarjam, A. Sarker, K. OConnor, R. Ginn, and G. Gonzalez, "Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features," *Journal of the American Medical Informatics Association*, p. ocu041, 2015.
[42] O. Bodenreider, "The unified medical language system (umls): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl 1, pp. D267–D270, 2004.
[43] F. Wang, P. Zhang, N. Cao, J. Hu, and R. Sorrentino, "Exploring the associations between drug side-effects and therapeutic indications," *Journal of biomedical informatics*, vol. 51, pp. 15–23, 2014.
[44] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.
[45] "RxList: The Internet Drug Index," http://www.rxlist.com/.
[46] "Drugs.com," http://www.drugs.com/.
[47] "Medicalook," http://www.medicalook.com/.
[48] "Pharmacy and drugs," http://www.pharmacy-and-drugs.com/.