DataRank: An Online Ranking Algorithm for Ranking Biomedical Datasets

Firstname A. Lastname, MD, MPH[1], Firstname B. Lastname, MD, PhD[2]
[1] Institution, City, MA; [2]Institution, City, CA

Abstract

In this paper, we propose an online ranking algorithm, DataRank for ranking biomedical datasets that are used in the papers indexed in PubMed Central. DataRank takes the bipartite citation graph between datasets and articles and dataset features, i.e. MeSH terms, as input and rank datasets according to their relevance to the searching query as well as user feedbacks. DataRank constituted dataset features by aggregating MeSH terms from the connected papers in the bipartite graph. For each search query, DataRank first maps the "free text query" to a MeSHQuery and yield an offline *ranking of datasets for the MeSHQuery using a Bayesian approach which the likelihood is proportional to Jaccard index and prior is proportional to number of citations of that dataset. DataRank is also extended to a* online *algorithm by incorporating user-feedbacks regarding ranking relevance. The online DataRank again takes an Bayesian approach which uses offline DataRank as its prior and computes its likelihood by estimating the user rating for unknown values using collaborative filtering. A demo web search engine has been developed to rank more than 20,000 dataset that has been discovered in more than 1 million papers.*

Introduction

Over the past decades, a wide range of datasets including clinical, gnomic, imaging, behavioral, etc. is created in biomedical research, and there is no unified and systematic way for retrieving them. Although several repositories(e.g. DBGap, TCGA, GEO, etc.) has been established[?], unfortunately like many cases of big-data applications with a "diminishing return", retrieval of datasets is getting harder as the number of repositories and active datasets increases. In fact, the current indexing and searching system suffers from a number of problems, including

(I) **Indexing** of datasets rely on human resources, which is a costly and noisy precess.

(II) **Searching** between and within the repositories is not well developed, and user should know the dataset and repository prior to search.

(III) **Integration** of the repositories is widely ignored and indexing, searching and maintenance of each repository is being done independently.

(IV) **Privacy**, only contain public datasets, which are only a small fraction of all the datasets is being generated. (Not sure, should check it out later)

(V) **Ranking** of search results of each repository is being done naively based matching of the query to the dataset's limited metadata.

(VI) **Recommendation** is now is a part of most of modern information retrieval systems with large amount of users and items, but it has not been considered for datasets and research scientists.

In this paper, we provide a solution for problems (I)-(VI), respectively by

(i) **Crawling** the PMC articles for patterns and rules provided by NIH to discover datasets directly from papers.

(ii) **Integrating** all the discovered datasets into one unified repository.

(iii) **Providing**, general information for both private and public datasets.

(iv) **Ranking** datasets using different informative features including MeSH, number of citations, etc.

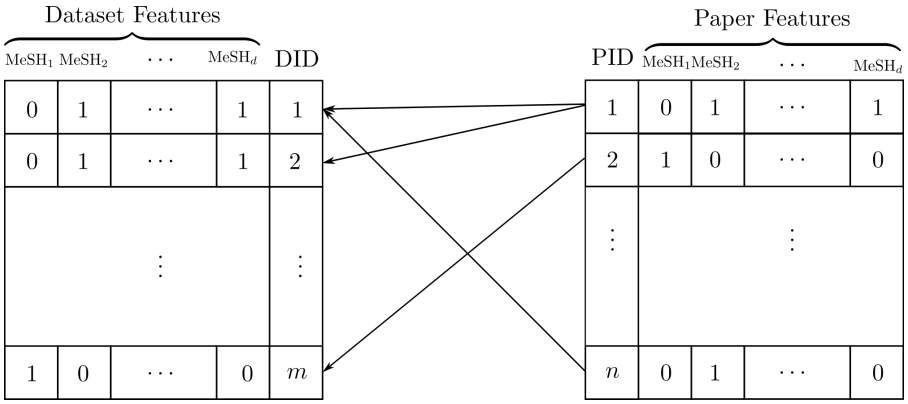

Figure 1: A bipartite graph between m datasets and n papers with MeSH terms as their features. PID and DID are Paper ID and Dataset ID respectively where existence of an edge between a paper and a dataset implies that dataset is used in that paper.

(v) **Recommendation** of other datasets beyond current search results to make related datasets even more discoverable.

It should be noted that the task of crawling and indexing datasets is a non-trivial and independent task to others, so the methods and techniques for implementing it are outside of the scope of this manuscript. Therefore, in the rest of this paper we mainly focus on developing efficient methods for (ii)-(v) and for the crawling task we take an simple method as follows: By creating regular expressions for all the citation rules provided by NIH for each repository we discovered more than 20,000 datasets out of more than 1 million PMC full text articles and created a bipartite graph between dataset ids and paper ids. The product of crawling process is actually a bipartite graph, though incomplete, between articles and datasets. In the rest of this paper we describe our methodology to process this bipartite graph, Figure **??**.

The rest of this paper is organized as follows: in section 1 we first review existing methods for enhancing information retrieval systems and then in section 2 we present out methods for the aforementioned problems. We outline implementation remarks and experimental study in section 6 and finally we make conclusions and state possible future works in section 9.

1 Background

- information retrieval
- Pubmed
- MeSH
- ranking
- recommendation

2 Methodology

The DataRank algorithm is an online algorithm which updated its predictive model after receiving explicit and implicit feedbacks from users. Specifically, we resort to incorporate user feedbacks explicitly you giving an option to the user to rate the search results, and we interpret user clicks as implicit feedbacks. Therefore, for every new user, the algorithm works in an offline manner and after receiving user feedbacks it updates the model to present user specific rankings. In the following parts we first describe the features and then explain our DataRank model offline ranking and finally extend offline DataRank to online setting.

3 Features

In this paper we use the corresponding set of MeSH terms as features for each articles and use the bag-of-words representation for representing documents. More precisely, the corpus n articles is represented by a $d \times n$ binary matrix, $M \in \{0,1\}^{d \times n}$. Similarly we define the matrix of $X \in \{0,1\}^{d \times m}$ for representing m dataset via bag-of-words of MeSH terms, where $\mathbf{x}_i$ is the i^{th} column of X, is i^{th} document in the corpus. Also, for each dataset, the dataset label $\mathbf{y}_i$, is the dataset identifier which is a categorical variable, i.e., $\mathbf{y} \in \{1 \cdots m\}^m$, the bipartite graph is represented via the adjacency matrix $A \in \{0,1\}^{n \times m}$. Since we are considering binary features for both documents and datasets, the dataset features can be readily obtained from the document features and adjacency matrix:

$$X = \min(MA, 1) \tag{1}$$

where $\min(\cdot)$ is a elementwise operator over its matrix argument.

4 Offline Ranking

For the problem of ranking we consider a probabilistic approach and propose a graphical model to specify dependencies between random variables in the model. Therefore, $\mathbf{x}, \mathbf{y}$ should henceforth be understood as realizations of the corresponding random variables. We also introduce another random variable $\mathbf{q}$ for search queries over the same sample space as $\mathbf{x}$, i.e., MeSH terms which $\mathbf{q} \in \{0,1\}^d$. Finally, $\mathbf{c}$ is an (observed) random variable which defines a prior over the labels $\mathbf{y}$. The graphical model shown in Figure 2-(a). It should be noted that, in any case variables $\mathbf{x}$ (dataset features) and $\mathbf{c}$ (datasets prior) are observed and we never need their marginal distributions and therefore we only consider the query $\mathbf{q}$ as evidence.

In this model, the problems of ranking for a given query $\mathbf{q}$ is to find the posterior distribution for the dataset labels given evidence. More precisely the posterior distribution

$$\Pr(\mathbf{y}|\mathbf{q}, \mathbf{c}, \mathbf{x}) \propto \Pr(\mathbf{y}|\mathbf{c}) \Pr(\mathbf{q}|\mathbf{x}) \tag{2}$$

full specifies the ranking, where the posterior distribution is expanded according to the graphical model Figure 2-(a). Since, $\Pr(\mathbf{y}|\mathbf{c})$ is not function of evidence, we can legitimately consider it as dataset-prior and consider $\Pr(\mathbf{q}|X)$ as query-likelihood.

In the standard statistical modelling procedures, the next step is to specify dataset-prior and query-likelihood distributions.

Query-likelihood. Since the binary feature vector $\mathbf{x}$ and query $\mathbf{q}$ are representation of sets, using Vann diagram, we can easily compute the likelihood

$$\Pr(\mathbf{q}|\mathbf{x}) = \frac{\Pr(\mathbf{q}, \mathbf{x})}{\Pr(\mathbf{x})} = \frac{|\mathbf{q} \cap \mathbf{x}|}{|\mathbf{x}|} \tag{3}$$

where set operations are applied to the set representation of the binary vectors.

However, this probability does not account for mismatches of terms in the query and features, and this leads to the phenomena that two queries whit the same number of matches but different number of mismatches have the same probability. To remedy this crucial problem, we can add the number of mismatches to the denominator, which is equivalent to condition on $\mathbf{x} \cup \mathbf{q}$

$$\widehat{\Pr}(\mathbf{q}|\mathbf{x}) \propto \Pr(\mathbf{q}|\mathbf{x} \cup \mathbf{q}) = \frac{|\mathbf{q} \cap \mathbf{x}|}{|\mathbf{q} \cup \mathbf{x}|} \tag{4}$$

the value of (5) is known as Jaccard index or Tanimoto or Jaccard coefficient[?] . Thus for the query likelihood we have

$$\widehat{\Pr}(\mathbf{q}|\mathbf{x}) = \frac{\widehat{\Pr}(\mathbf{q}|\mathbf{x})}{\sum_{i=1}^{n} \widehat{\Pr}(\mathbf{q}|\mathbf{x}_i)} \tag{5}$$

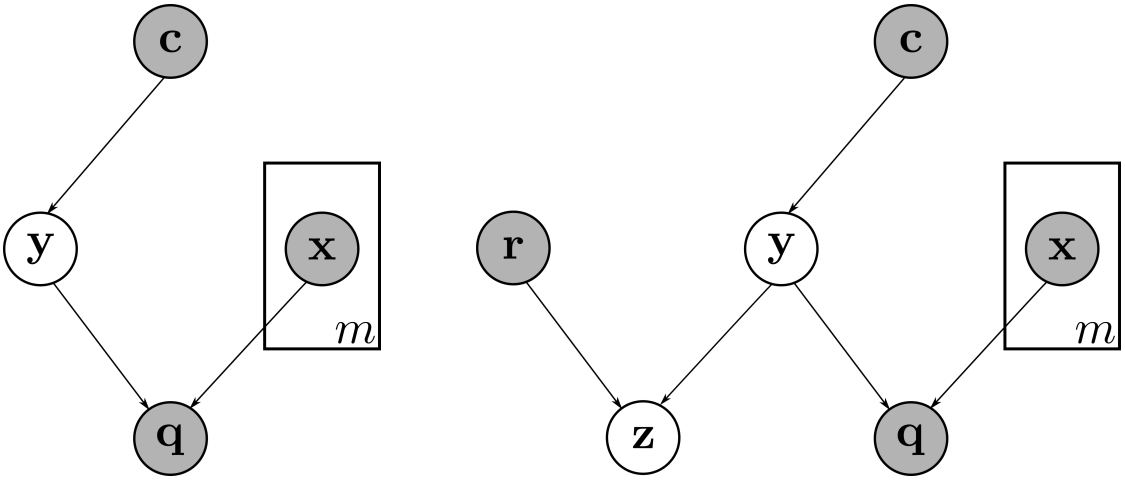

Figure 2: Probabilistic graphical models for offline (a) and online (b) methods for ranking datasets. The shaded nodes are observed variable, arrow implies conditional dependence and rectangles are short hand for replication of the inside nodes. For details see text.

Dataset-prior. Instead of learning a prior for datasets using empirical Bayesian methods, we simply chose to propose a reasonable subjective prior for the datasets. Basically, the better way to think about prior is to imagine the posterior distribution in a scenario where there is no evidence available, which is equivalent to say that what is the best ranking of datasets if do not have a query. There are many ways to answer this question, but a reasonable approach is to sort them based on their general popularity. Interestingly, we can quantitatively specify the popularity for the datasets by taking into account of how many citations they have in the training dataset, i.e., $\mathbf{c} = \mathbf{1}^T A$. More precisely, we can write prior as

$$\Pr(\mathbf{y}|\mathbf{c}) = \frac{\mathbf{c}}{\mathbf{c}^T\mathbf{1}} = \frac{\mathbf{1}^T A}{\mathbf{1}^T A \mathbf{1}} \tag{6}$$

where $\mathbf{1}$ is the vector of ones of corresponding dimension. So far, we have fully specified the prior and likelihood and hence posterior for the offline model and the ranking of datasets amounts to merely sort datasets decreasingly according to their posterior distribution.

5 Online Ranking

In this part we extend the offline-DataRank to the online setting by incorporating user feedback into ranking. For this porous, we propose a new model, Figure 2-(b), which introduces *incomplete* user ratings $\mathbf{r} \in \{0 \cdots k\}^m$, *completed* user rating $\mathbf{R} \in \mathbb{R}_+^m$ and the dataset online-labels $\mathbf{z}$, where k is the number of different state of ratings for each result in the ranking and 0 is used to denote unknown values in the ratings. As shown in the graphical model, the online-labels depend on user feedback but, (offline) dataset labels $\mathbf{y}$ do not.

The ratings $\mathbf{r}$ are initialized with zero value and at each epoch users rates the search results and updates the $\mathbf{r}_t$. Similar to offline method, the task is to compute the posterior

$$\Pr(\mathbf{z}|\mathbf{r}, \mathbf{y}, \mathbf{c}, \mathbf{q}, \mathbf{x}) \propto \Pr(\mathbf{z}|\mathbf{r}) \Pr(\mathbf{y}|\mathbf{c}, \mathbf{q}, \mathbf{x}) \tag{7}$$

where the factorization induced by the graphical model Figure 2-(b) and in this model evidence is incomplete user rating $\mathbf{r}$. Interestingly the offline posterior $\Pr(\mathbf{y}|\mathbf{c}, \mathbf{q}, \mathbf{x})$ can be regarded as prior in this model, which implies that without any evidence, user ratings, the online posterior is exactly equal to its prior, i.e., offline posterior, which makes a perfect sense.

Specifying the likelihood $\Pr(\mathbf{z}|\mathbf{r})$ amounts to determining unknown values in the rating vectors and then normalize it.

We use collaborative filtering[?] to estimate unknown values of user ratings for the datasets that has not been rated yet, based on similarity of rated datasets a vector completion method, to use the sparse and incomplete past ratings to fill the unknown values based on similarity of datasets to the unrated datasets and the all the legitimate ratings. Thus, the key step is to define a similarity measure between datasets. Regarding, the binary MeSH representation of datasets, we opt to use Tanimoto kernel[?] between datasets for measuring similarity between datasets

$$K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{k}_\cap(\mathbf{x}_i, \mathbf{x}_j)}{\mathbf{k}_\cap(\mathbf{x}_i, \mathbf{x}_i) + \mathbf{k}_\cap(\mathbf{x}_j, \mathbf{x}_j) - \mathbf{k}_\cap(\mathbf{x}_i, \mathbf{x}_j)} \tag{8}$$

where K is a $m \times m$ symmetric positive definite matrix, $\mathbf{k}_\cap(\mathbf{x}_i, \mathbf{x}_j) = \frac{|\phi(\mathbf{x}_i) \cap \phi(\mathbf{x}_j)|}{|\phi(\mathbf{x}_i) \cup \phi(\mathbf{x}_j)|}$ is intersection kernel[?] between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\phi(\cdot)$ is a general nonlinear feature function.

Having a similarity matrix between datasets, using collaborative filtering we can readily fill the unknown values in the rating vector, and compute the likelihood

$$\Pr(\mathbf{z}|\mathbf{r}) = \frac{K\mathbf{r}}{\mathbf{1}^T K \mathbf{r}} \tag{9}$$

Using (5) and (9) we can fully specify the online prediction posterior (7) and therefore, online ranking for the DataRank algorithm.

6 Experimental Study

7 Implementation

8 Experiments

9 Conclusions

References