QAPhenolyzer: A Phenotype Based Causal Gene Analyzer with Embedded Question Answering System

Cong Liu
An HPO term based simulation study

February 6, 2018

1 Problem Statement

Phenolyzer could use observed phenotypes to infer the genotype in a given patient. However, in most genetic counseling scenario, a genetic counselor is first provided an incomplete description of a patient's phenotypes. In order to reach a further conclusion, the counselor has to ask or exam more phenotypes to obtain more information. This counseling procedure is a dynamic procedure rather than a one-shot answer. In this study, we design a QAPhenolyzer to mimic the counseling procedure. Given a set of initial phenotype list, a prior distribution is calculated, and then for each candidate missing phenotype, we calculate the information gain based on expected *Kullback–Leibler* divergence. And then we select the top ranked phenotypes to exam. The procedure is repeated until we get satisfied result.

Next, we first implement a lite version of Phenolyzer in *section 2*. Then we describe how to select top ranked feature in *section 3*. The QA system pipeline is described in *section 4*. A simulation results are showed in *section 5*. *Section 6* lists the future plan.

2 Knowledge Encoder

HPO database provides two files to link between genes and HPO-terms. The knowledge is represented as a collection of tuples (gene, hpo term). We designed a probabilistic model to re-encode the knowledge. For a given individual, we use a vector X to represent the status of phenotypes, where

$$x_i = \begin{cases} 1, & i \text{ is observed.} \\ -1, & i \text{ is not observed.} \\ 0, & \text{information on } i \text{ is missing.} \end{cases} \tag{1}$$

Let y be the outcome of patient, and we used a softmax function to estimate the probability $P(Y = y_j|X)$

$$P(y_j|X) = \frac{e^{z^{(j)}}}{\sum_{j \in J} e^{z^{(j)}}} \tag{2}$$

where we define the net input z as

$$z^{(j)} = \sum_i w_i^{(j)} x_i \tag{3}$$

and

$$w_i^{(j)} = \begin{cases} 1, & (j,i) \text{ exists in knowledge base.} \\ -1, & \text{o/w.} \end{cases} \tag{4}$$

Notice that -1 represents a negative rule. We set $w_i^{(j)}$ as 0 if we don't have any knowledge. However, since in most cases, we could assume the phenotype is not related to a gene. In this study, we will only use 1 and -1 to represent our knowledge. In general, We could manually change $w^{(j)}$ based on expert annotation or estimated from a training dataset.

3 Information Gain

We use *Kullback–Leibler* divergence to measure the information gain. The Kullback–Leibler divergence is a measure of how one probability distribution diverges from a second, expected probability distribution. It is defined as

$$D_{KL}(P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \tag{5}$$

In Bayesian statistics the KullbackLeibler divergence can be used as a measure of the information gain in moving from a prior distribution to a posterior distribution. Given some extra information on x_t, the information gain is

$$D_{KL}(P; J, I, x_t) = \sum_{j \in J} P(y_j|I, x_t) log \frac{P(y_j|I, x_t)}{P(y_j|I)} \tag{6}$$

where $P(y_j|I)$ is the prior distribution, and $P(y_j|I, x_t)$ is the posterior distribution after the extra information of x_t is provided. For each candidate missing phenotype t, the expected *Kullback–Leibler* divergence $E_{x_t}[D_{KL}(P||Q)]$ is used to scoring the informative level.

4 Interactive QA pipeline

Our PhenolyzerQA now could be implemented as a interactive question answering system. We start from a initial set of observed features and calculate a prior distribution. In each

step, we are trying to find the most informative feature and complete the value for this feature. Instead of calculate the "information gain" for all classes, we focus on the top ranked classes given the prior distribution. Therefore, in each step, we reduce the number of candidate classes, and calculate the probability conditioning on the response is one of the candidate classes. After identify the most informative feature, we fill out the missing value and update the prior distribution. The procedure will continued until the number of questions raised are reach the maximum.

Algorithm 1 Interactive QA System

Inputs:
K: the number of candidate gene left in first step.
M: the total number of questions.
$I = \{x_i\}^{(0)}$: initial feature set.
$W = \{w_i^{(j)}\}$: feature weight matrix.
$P(y_j|X, W)$: probability function for y_j
Initialize:
$k \leftarrow K$
$q = P(y_j|I)$
$r = int(M/K)$
while $k > 0$ **do**
  $J \leftarrow \{j : y_j$ in the first k ordered by $q\}$
  $T \leftarrow \{i : \exists(j, j') \in J, w_i^{(j)} \neq w_i^{(j')}\}$
  $t = argmax \sum_{t \in T} E_{x_t}[D_{KL}(P; J, I, x_t)]$
  Exam the feature value of x_t (Question-Answer)
  $I \leftarrow I + \{x_t\}$
  $q = P(y_j|X; j \in J)$
  $k \leftarrow k - r$
end while

5 A simulation results

We performed a simulation study to evaluate the performance of our algorithm. For each simulation, We randomly sample 10 observed features, and use the encoder method described in *section 1* to generate the class distribution. The true classes is randomly sampled from the top 50 classes with the corresponding probability. Here the values of 10 features are all 1 (i.e. observed) due to the nature of clinical practice. We set $K = 1$ and $M = 5$ to perform the interactive QA system. To evaluate the performance of our algorithm we compare it with the random selected features. Same number of missing features ($M = 5$) are randomly selected to exam. To make the comparison fair, we only select missing features whose w_j's are not the same across all top 50 classes. In addition, we compare the interactive feature selection with global feature selection. In global feature selection, the top 5 features are selected according to the information gain comparing with the initial 10 observed features. The results are summarized in Table 1.

Statistics	Interactive-QA	Global-QA	Random-QA	Init
1st-quantile	1.00	1.25	6.00	8.25
median	2.00	4.00	15.0	18.0
3rd-quantile	4.00	9.75	28.0	43.5
mean	2.80	7.30	25.3	30.1
sd	1.98	8.21	27.9	29.3

Table 1: Comparison between different methods

6 Future work

- The algorithm should be evaluate in a real-world dataset. Use CKD for example.
- A more comprehensive simulation should be performed with different parameters.
- Other methods should be compared. For example, compare different IG method.
- User interface should be implemented.
- the phenolyzer lite should be upgraded by considering ontology, other knowledge base and gene-gene interaction. Unfortunately, if it is very difficulty to design a probabilistic model based on current phenolyzer program.