Ontological Analysis

Young-Rae Cho
Associate Professor
Department of Computer Science
Baylor University

Ontology

- **Ontology in Philosophy**
  - The study of the nature of being or existence including their categories and their relations (wikipedia)

- **Ontology in Computer Science**
  - The specification of a conceptualization: description of the concepts and relationships that exist for an agent or a community of agents (Gruber)
  - A set of representational primitives (i.e., classes, attributes, and relationships) for modeling a domain of knowledge

- **Ontology in Biology and Bioinformatics**
  - A formal way of representing biological knowledge which is described by the concepts and their relationships to each other (Bard and Rhee)
Representations of Ontology

- **Components**
  - Concepts and Relationships

- **Representation**
  - Graph (concepts → nodes, relationships → edges)

## Relationships in Ontology

- **Directions**
  - Relationships are generally directed
  - Concepts have parent-child relationships

- **Properties** (in tree or DAG)
  - Antisymmetric
  - Transitivity

- **Examples**
  - "is-a" relationship
  - "part-of" relationship
MIPS Functional Catalogue

- **Features**
  - The functional concepts are nodes
  - Tree structure for the relationships between functional concepts
  - Provides functional annotation for small model organisms (functional descriptions of genes or proteins)
  - Manually created
  - Well-suited for the annotation of genome from different domains of life
  - [http://mips.helmholtz-muenchen.de/funcatDB/](http://mips.helmholtz-muenchen.de/funcatDB/)

Gene Ontology (GO)

- **Features**
  - Organized by GO Consortium
  - A repository of bio-ontology (controlled vocabularies) databases - consistent descriptions across different organisms
  - Nodes represent GO terms structured in 3 main categories: biological processes, molecular functions, and cellular components
  - DAG for the relationships between GO terms
  - Provides annotation of genes and gene products
  - Created by any published evidence (mostly, from high-throughput data)
  - Data curation, e.g., redundant annotation elimination
GO Structure

- Example of GO DAG structure

Human Phenotype Ontology (HPO)

- Features
  - A repository of phenotypic information of human
  - Nodes represent the HPO terms describing phenotypic features
  - DAG for the relationships between HPO terms
  - Provides annotation of human genes and gene products
  - Based on OMIM, a catalog of human genes and genetic disorders
  - Data manual curation

  - http://human-phenotype-ontology.github.io/
Research Topic 1. Semantic Similarity Analysis

➤ Definition of Semantic Similarity
   ▪ Ontological relatedness between two concepts
   ▪ In Gene Ontology, similarity between two terms

➤ Categories
   ▪ Ontology structure-based methods
     • Edge-based methods
     • Node-based methods
   ▪ Annotation-based methods
   ▪ Integrative methods

Edge-Based Measures

➤ Path length between two terms
   \[ \text{sim}(C_1, C_2) = \frac{1}{\text{len}(C_1, C_2) + 1} \]

➤ Normalized path length between two terms by GO depth
   \[ \text{sim}(C_1, C_2) = -\log \left( \frac{\text{len}(C_1, C_2)}{2 \times \text{depth}} \right) \]

➤ Depth to the most specific common ancestor

➤ Normalized depth to the most specific common ancestor by average depth to two terms
   \[ \text{sim}(C_1, C_2) = \frac{2 \times \text{len}(C_{root}, C_0)}{\text{len}(C_0, C_1) + \text{len}(C_0, C_2) + 2 \times \text{len}(C_{root}, C_0)} \]

where \( C_0 \) is the most specific common ancestor term
Node-Based Measures

- Number of common ancestors
  \[ \text{sim}(C_1, C_2) = |P(C_1) \cap P(C_2)| \]
  where \( P(C) \) is the set of ancestors of the term \( C \)

- Normalized number of common ancestors
  - Jaccard index
    \[ \text{sim}(C_1, C_2) = \frac{|P(C_1) \cap P(C_2)|}{|P(C_1) \cup P(C_2)|} \]
  - Dice index
    \[ \text{sim}(C_1, C_2) = \frac{2 \times |P(C_1) \cap P(C_2)|}{|P(C_1)| + |P(C_2)|} \]
  - Min normalization
    \[ \text{sim}(C_1, C_2) = \frac{|P(C_1) \cap P(C_2)|}{\min(|P(C_1)|, |P(C_2)|)} \]

Information Contents

- Formulation
  - In Information Theory, the information content of a concept \( C \) is defined as
    \[ -\log P(C) \]

- Transitivity Property of Annotations
  - If a gene \( g \) is annotated to a term \( C \), then it is also annotated to all the ancestor terms of \( C \) towards the root
  - The likelihood of \( C \) can be defined by the annotation on \( C \)
    \[ P(C) = \frac{\text{the number of genes annotated to } C}{\text{the number of all genes annotated to the ontology}} \]
Annotation-Based Measures

- Information content of the most specific common ancestor
  \[ \text{sim}(C_1, C_2) = - \log P(C_0) \]
  where \( C_0 \) is the most specific common ancestor

- Normalized information content of the most specific common ancestor by average information content of two terms
  \[ \text{sim}(C_1, C_2) = \frac{2 \times \log P(C_0)}{\log P(C_1) + \log P(C_2)} \]

- Sum of differences between information content of the most specific common ancestor and information content of two terms
  \[ \text{sim}(C_1, C_2) = \frac{1}{2 \log P(C_0) - \log P(C_1) - \log P(C_2) + 1} \]

Integrative Methods

- Combination of an edge-based measure and a node-based measure
  \[ \text{sim}(C_1, C_2) = \sum_{C_0 \in \{P(C_1), P(C_2)\}} \text{len}(C_{\text{root}}, C_0) \]

- Combination of a node-based measure and an annotation-based measure
  \[ \text{sim}(C_1, C_2) = \frac{\sum_{C_0 \in \{P(C_1), P(C_2)\}} \log P(C_1)}{\sum_{C_0 \in \{P(C_1), P(C_2)\}} \log P(C_0)} \]

- Combination of two annotation-based measures
Problems of Semantic Similarity

- **Node-Based Methods**
  - Assumes that all GO terms are meaningful
    (Terms have been randomly created based on evidence.)

- **Edge-Based Methods**
  - Assumes that all relationships represent the same quantity of similarity
    (Relationships have been randomly created based on evidence.)

- **Annotation-Based Methods**
  - Applicable only if genes are fully annotated

Applications of Semantic Similarity

- **Applications**
  - Functional prediction of incompletely annotating genes
  - Semantic similarity between terms (concepts)
    → Functional similarity between genes

- **Challenges**
  - A single gene performs multiple functions
  - A single gene is annotated on multiple terms
    - $X = \{X_1, X_2, \ldots, X_m\}$ are the most specific terms having a gene $g_1$
    - $Y = \{Y_1, Y_2, \ldots, Y_n\}$ are the most specific terms having a gene $g_2$
Implementation of Functional Similarity

- Functional Similarity between Genes
  - Measuring semantic similarity between two sets of terms

- Pairwise Methods
  - Measuring semantic similarity between terms
  - Aggregating term-to-term semantic similarities
  - Ex, edge-based semantic similarity methods

- Group-wise Methods
  - Measuring semantic similarity directly between two sets of terms
  - Ex, node-based semantic similarity methods

Aggregation of Semantic Similarities

- Pairwise Averaging
  - Average of semantic similarity scores between $X_i$ and $Y_i$
  \[
  sim(g_1, g_2) = \frac{\sum_{i,j} sim(X_i, Y_j)}{|X| \times |Y|}
  \]

- Best Matching
  - Maximum semantic similarity score between $X_i$ and $Y_i$
  \[
  sim(g_1, g_2) = \max_{i,j} sim(X_i, Y_j)
  \]

- Best-Match Averaging
  \[
  sim(g_1, g_2) = \frac{\sum_{i} \max_{j} sim(X_i, Y_j) + \sum_{j} \max_{i} sim(X_i, Y_j)}{|X| + |Y|}
  \]
Research Topic 2. Association Rule Analysis

- Definition of Association Rules
  - One-directional relationship between two sets of items
  - In Gene Ontology, association rules from a term A to a term B

- Categories
  - Pairwise association rules vs. Multi-terms association rules
  - Within-ontology rules vs. Cross-ontology rules

Association Rules Basics (1)

- Background
  - Used to solve a market basket problem
  - Found from a transaction database
    - Transaction: a set of items that are bought by one person at one time
    - Transaction database example

<table>
<thead>
<tr>
<th>T-ID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>bread, eggs, milk, diapers</td>
</tr>
<tr>
<td>2</td>
<td>coke, beer, nuts, diapers</td>
</tr>
<tr>
<td>3</td>
<td>eggs, juice, beer, nuts</td>
</tr>
<tr>
<td>4</td>
<td>milk, beer, nuts, diapers</td>
</tr>
<tr>
<td>5</td>
<td>milk, beer, diapers</td>
</tr>
</tbody>
</table>
Association Rules Basics (2)

- **Frequent itemsets**
  - A set if items (as a subset of any single transaction) which occur frequently across transaction
  - Itemset having support greater than (or equal to) a user-specified minimum support threshold

- **Support**
  - Frequency of a set of items across transactions

\[
support (A \rightarrow B) = P(A \cup B)
\]

Association Rules Basics (3)

- **Association Rules**
  - One-directional relationship between two sets of items (e.g., \( A \rightarrow B \))
  - Rules having confidence greater than (or equal to) a user-specified minimum confidence threshold

- **Confidence**
  - For \( A \rightarrow B \), percentage of transactions containing \( A \) that also contain \( B \)

\[
confidence (A \rightarrow B) = P(B \mid A) = \frac{P(A \cup B)}{P(A)}
\]
Association Rule Mining

- **Process**
  1. Find frequent itemsets
  2. Find association rules

- **Brute Force Algorithm**
  - Enumerate all possible subsets of the total itemset
  - Count frequency of each subset
  - Select frequent itemsets

- **Apriori Algorithm**
  - Iterative increment of the itemset size
    1. Candidate itemset generation
    2. Frequent itemset generation

Apriori Algorithm Details

- **Downward Closure Property**
  - Any superset of an itemset X cannot have higher support than X
  - If an itemset X is frequent (support of X is higher than minimum support), then any subset of X must be frequent

- **Candidate Itemset Generation**
  - Selective joining
    - Each candidate itemset with size k is generated by joining two frequent itemsets with size (k-1)
    - The frequent itemsets with size (k-1) which share a frequent sub-itemset with size (k-2) are joined
  - A priori pruning
    - A frequent itemset with size k which has any infrequent sub-itemsets with size (k-1) is pruned
Applications of Association Rules

- Applications
  - Application to GO annotation data
    
    | Gene ID | GO terms                      |
    |---------|-------------------------------|
    | 1       | Term-1, Term-4, Term-6        |
    | 2       | Term-2, Term-4                |
    | 3       | Term-1, Term-3, Term-4        |
    | 4       | Term-3, Term-5, Term-6, Term-7|

  - Curation of GO (by within-ontology rules)
  - Functional prediction of incompletely annotating genes
  - Correction of inconsistent annotations

Questions?

- Lecture Slides are found on the Course Website,
  web ecs baylor edu faculty cho 5330