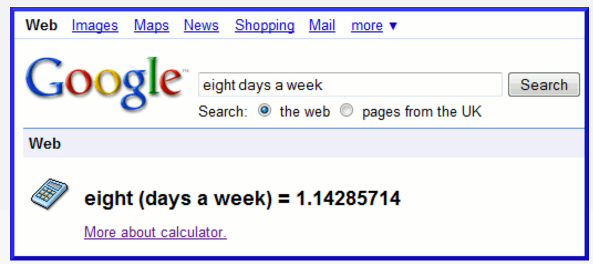
VIRTUAL ASTRONOMICAL OBSERVATORY

Semantics and Data Mining
or
How to decide what is useful

Matthew J. Graham, Caltech

What use is semantics in KDD?

- Data mining is “the *semi-automatic* discovery of patterns, associations, changes, anomalies, and statistically significant structures and events in data”
- Such discoveries are **evaluated** (filtered) based on **relevance** (according to some metric of interestingness) and **content** (qualitative condition based on domain knowledge) constraints
- Traditionally the user assumes the responsibility of choosing which aspects of the domain knowledge are most important for the current task (hence *semi-automatic*)

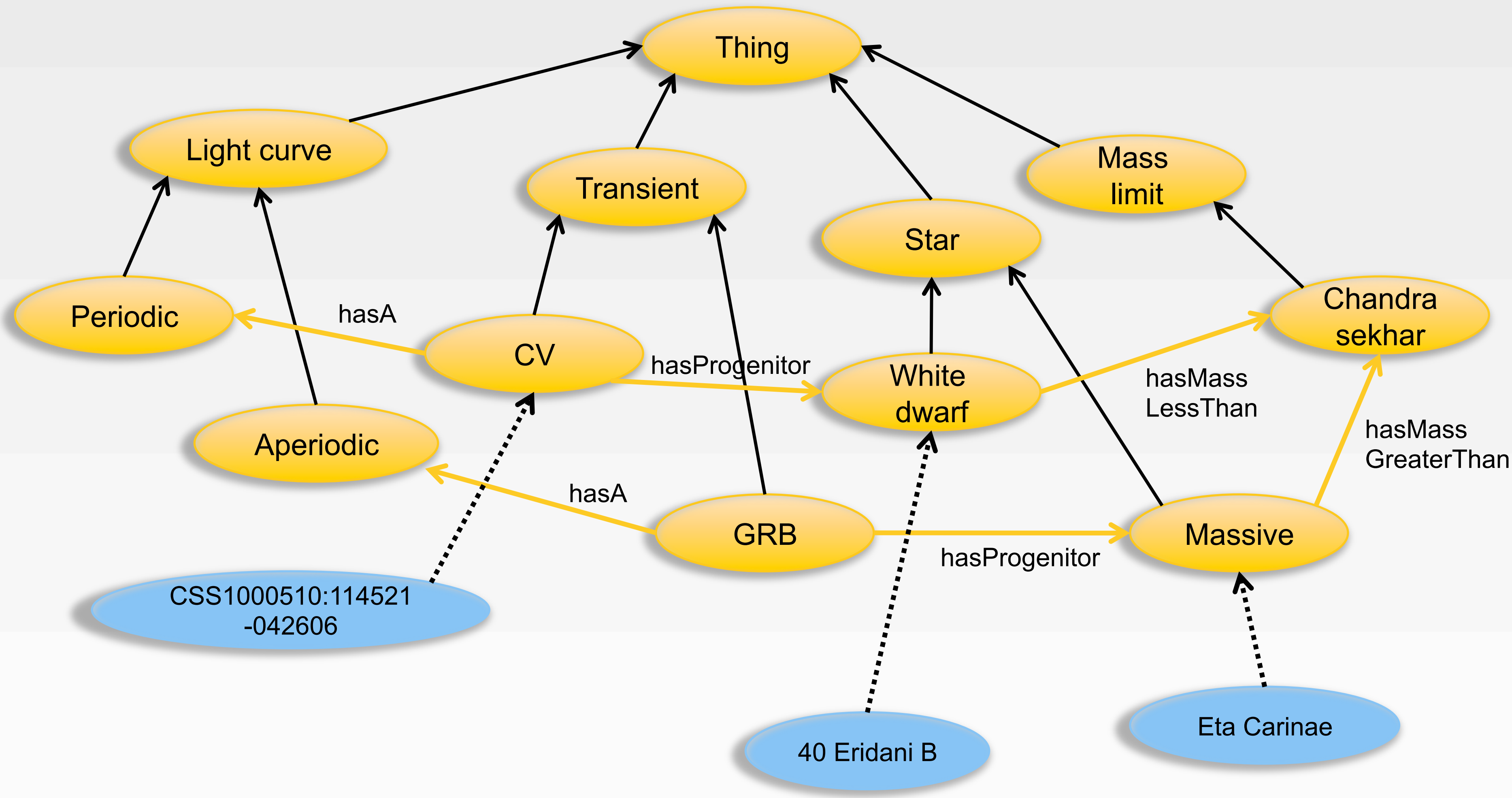


- One of the ten challenging problems in data mining research is the incorporation of background or domain knowledge into the discovery process (Yang & Wu 2006)
- The main difficulty lies in representing and acquiring domain knowledge
- Ontologies are a viable construct for representing knowledge (OWL, SWRL, SPARQL/SQRWL)

Definitions

- An **ontology** is a specification of an abstract, simplified view of a domain: it is a 5-tuple $o := [\mathcal{C}, \mathcal{R}, \mathcal{H}^c, rel, \mathcal{A}^o]$
  - $\mathcal{C}$ is a set of concepts which represent the entities in the ontology domain
  - $\mathcal{R}$ is a set of relations defined among concepts
  - $\mathcal{H}^c$ is a taxonomy which defines *is-a* relations among concepts
  - *rel* is a function that specifies the relations on $\mathcal{R}$ such that if *r* belongs to $\mathcal{R}$, $rel(r) = (c_1, c_2)$
  - $\mathcal{A}^o$ is a set of axioms that describe constraints on the ontology expliciting implicit facts
- A **knowledge base** specifies an instantiation for a particular ontology: it is a 4-tuple KB := [o, $\mathcal{I}$, *inst*, *instr*]
  - o is an ontology
  - $\mathcal{I}$ is a set of instances
  - *inst* is the concept instantiation function mapping $\mathcal{C}$ to $2^{\mathcal{I}}$
  - *instr* is the relation instantiation function mapping $\mathcal{R}$ to $2^{\mathcal{I}x\mathcal{I}}$

Example - I


Thing
Light curve
Transient
Star
Mass limit
Periodic
hasA
CV
hasProgenitor
White dwarf
Chandra sekhar
hasMass LessThan
hasMass GreaterThan
Aperiodic
hasA
GRB
hasProgenitor
Massive
CSS1000510:114521 -042606
40 Eridani B
Eta Carinae

Example - II

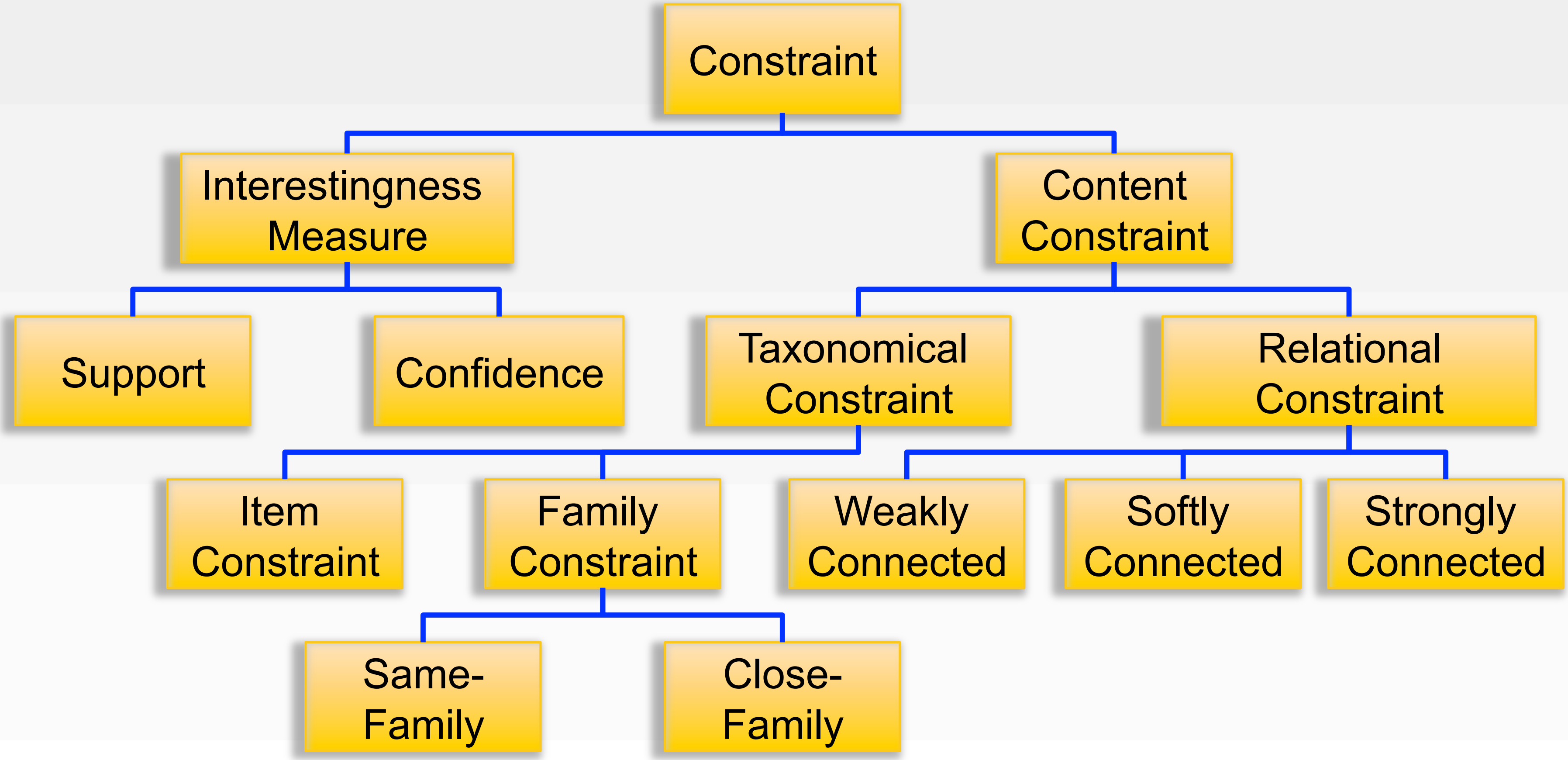
- **Ontology**
  - $\mathcal{C}$:= {Thing, Light curve, Transient, Star, Mass limit, Periodic, Aperiodic, CV, GRB, White dwarf, Massive, Chandrasekhar}
  - $\mathcal{R}$:= {hasProgenitor, hasA, hasMassLessThan, has MassGreaterThan}
  - $\mathcal{H}^c$:= {(Light curve, Thing), (Periodic, Light curve), (Aperiodic, Light curve), (Transient, Thing), (CV, Transient), (GRB, Transient), (Star, Thing), (White dwarf, Star), (Massive, Star), (Mass limit, Thing), (Chandrasekhar, Mass limit)}
  - *rel*: hasProgenitor(CV, White dwarf), hasProgenitor(GRB, Massive), hasA (CV, Periodic), hasA(GRB, Aperiodic), hasMassLessThan(White dwarf, Chandrasekhar), hasMassGreaterThan(GRB, Chandrasekhar)
- **Knowledge base**
  - *o* := [$\mathcal{C}$, $\mathcal{R}$, $\mathcal{H}^c$, *rel*, $\mathcal{A}^o$:={}]
  - $\mathcal{I}$:= {CSS1000510:114521-042606, 40 Eridani B, Eta Carinae}
  - *inst* := {(CSS100510:114521-042606, CV), (40 Eridani B, White dwarf), (Eta Carinae, Massive)}

Application ontologies

- Contains essential knowledge in order to drive data mining tasks
- Smart workflows
  - Recommender systems
  - Competitive intelligence tools
- OntoDM (http://kt.ijs.si/panovp/OntoDM):
  - dataset: data items
  - datatype: primitive, structured
  - data mining task: predictive modelling, pattern discovery, clustering, probability distribution estimation
  - generalization: predictive model, pattern, clustering, probability distribution
  - data mining algorithm: distance function, kernel function, refinement operator
  - function: aggregation function, prototype function, evaluation function, cost function
  - constraint: evaluation, language constraint
  - data mining scenario: query, inductive query

Incorporating ontologies

- A simple way to incorporate an ontology into a data mining process is as a filter to prune those discoveries that do not meet the imposed constraint (derived from the ontology)



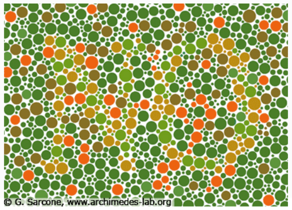
Constraints

- A **constraint** is a predicate on the power set of the set of items I, that is, it is a function *c*: $2^{\mathcal{I}}$ -> {true, false}. An itemset *S* is said to satisfy *c*, if and only if, *c*(*S*) is true.
- Interestingness metrics based on semantic similarity:
  - Edge counting: distance between ontology concepts
  - Information theoretic: information content of the lower common ancestor of two concepts

    $p_{ms}(c1,c2) = \min(\{p(c)\}$; $sim(c1, c2) = -\ln p_{ms}(c1, c2)$
- Taxonomical based on family ties
  - {White dwarf, Massive} have same parent
  - {White dwarf/DA, Massive} have common ancestor and are at least n^{th} (n=1) cousins to each other
- Relational based on relations between concepts
  - {Aperiodic, GRB, Massive} are weakly connected
  - No strongly connected itemsets

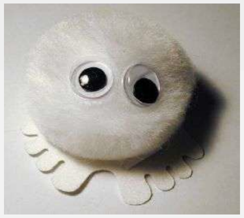
Data mining with ontologies - I

- **Clustering**:
  - Linkage-based:
    - the similarity between two objects is measured based on the similarities between the objects linked with them
  - Relational Fuzzy C-Means:
    - processes n vectors in p-space as data input, and uses them, in conjunction with first order necessary conditions for minimizing the FCM objective functional, to obtain estimates for two sets of unknowns
  - Correlation Cluster Validity
    - Validate number of clusters by computing correlation between reconstruction matrix after fuzzy clustering and original dissimilarity matrix
  - Ontological SOM
    - Represent contribution of ontology term to description of associated node and replace distance metric with an ontology-based dissimilarity measure



Data mining with ontologies - II

- **Detecting rare events via reasoning**
  - Application of description-logic reasoning over an ontology to automate classification of instances into family and subfamily groups
- **Fuzziness**
  - Markov Logic Networks – allows declarative domain knowledge to be expressed with real-valued weight indicating strength of statements



- **Association Rules**
  - Discover strong rules between concepts/instances using different measures of interestingness
- **Network characterization**
  - Establish functional relationships between instances and then predict functions and networks from these