Title
Constructing Hierarchical Concepts via Analogical Generalization

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 36(36)

ISSN
1069-7977

Authors
Liang, Chen
Forbus, Ken

Publication Date
2014

Peer reviewed

Constructing Hierarchical Concepts via Analogical Generalization

Chen Liang (chenliang2013@u.northwestern.edu)
Kenneth D. Forbus (forbus@northwestern.edu)
Qualitative Reasoning Group, Northwestern University, 2133 Sheridan Road
Evanston, IL, 60208 USA

Abstract

Learning hierarchical concepts is a central problem in cognitive science. This paper explores the *Nearest-Merge* algorithm for creating hierarchical clusters that can handle both feature-based and relational information, building on the SAGE model of analogical generalization. We describe its results on three data sets, showing that it provides reasonable fits with human data and comparable results to Bayesian models.

Keywords: Analogy, concept learning, computational modeling, hierarchical clustering

Introduction

How concepts are formed and organized is a central question in cognitive science. It has been argued that people group examples into categories to maximize within-category similarity and minimize between-category similarity (Mervis & Rosch, 1981). One important feature of categories is that they are not isolated from each other. Instead, people tend to organize the categories into a hierarchy or taxonomy (Collins & Quillian, 1969; Murphy & Lassaline, 1997).

Most models of hierarchical category learning focus on feature-based representations (e.g. Medin & Schaffer, 1978; Fischer 1987). Feature-based representations cannot capture relational thinking, involved in explanation, causal reasoning, and planning, which is central to human cognition (Gentner & Kurtz, 2005). Bayesian models (Kemp et al. 2006; Kemp & Tenenbaum 2008) can create a variety of structures, including hierarchical structures, although to our knowledge they have not been tested on representations involving higher-order relations. Analogical generalization (Kuehne et al. 2000) can handle relational representations with higher-order relations as well as feature representations, but currently it does not create hierarchical conceptual structures. This paper explores how analogical generalization might be extended to model the formation of hierarchical conceptual structure. The basic insight is that the numerical similarity score computed via structure-mapping can serve the same roles as vector-based operations used in feature-based models of similarity, and hence many of the same ideas and insights of those models can be extended to handle relational (including higher-order relational) representations.

We begin by summarizing relevant aspects of structure-mapping and the models that we are building upon. Then we describe the *Nearest-Merge* algorithms for constructing hierarchical concepts. Next we describe three experiments, providing evidence that it can produce results that are compatible with human judgments, and with a prior Bayesian simulation on a data set for which no human data is available. We close with related and future work.

Background

We assume Gentner's (1983) structure-mapping theory. Our model is built upon the Sequential Analogical Generalization Engine (SAGE; McLure et al. 2010), which in turn uses the Structure-Mapping Engine (Falkenhainer et al 1989) for analogical comparison and MAC/FAC (Forbus et al 1995) for analogical retrieval. Thus we start with SME, since it is the most fundamental. SME takes as input two structured representations, a base and target, and produces one or more mappings. Each mapping provides a set of correspondences (i.e. what goes with what), a structural evaluation score which provides an overall estimate of match quality, and candidate inferences. We refer to the similarity score of a mapping as NSIM(base,target), which is normalized to [0,1] by dividing the raw score by the mean of the self-scores of the base and target[1]. Forward candidate inferences go from base to target, reverse candidate inferences go from target to base. MAC/FAC takes as input a case library, which is a set of structured descriptions, and a probe, which is a structured description. It returns one or more approximations to the most similar case in the case library, using a two-stage process that enables it to scale to large case libraries. The first stage uses a flattened version of the relational structure of cases, called *content vectors*, whose dimensions are proportional to the weighted number of occurrences of each predicate in a description. The dot product of two content vectors is an estimate of SME's structural evaluation score for the structured representations, making it a useful coarse filter. Both SME and MAC/FAC have been used to model a variety of psychological phenomena.

SAGE maintains, for each concept, a *generalization context*. A generalization context has a trigger, which is used to test whether or not an incoming example should be added to it. (An incoming example might satisfy multiple triggers, and hence be processed by several generalization contexts.) Each generalization context maintains a set of *generalizations* and a set of *unassimilated examples*.

[1]The mapped representations are subsets of both base and target, so its score is lower than either of their self-scores.

(Either of these sets might be empty, and both are initially.) Generalizations are also structured representations, but associated with their statements are probabilities, based on the number of times facts that align with them are found in examples that are part of that generalization.

Every time a new example is added, SAGE uses MAC/FAC to retrieve up to three examples or generalizations, based on whatever is the most similar to the new example. If nothing is retrieved, or the similarity to the returned item is less than an *assimilation threshold*, the new example is stored as is. Otherwise, if the returned item is a generalization, the new example is assimilated into it. If the returned item is a previously unassimilated example, then the two are combined into a new generalization.

The assimilation process increments frequency counts associated with each statement, based on whether or not something in the example aligned with it. For a new generalization, such facts are always either 1.0 (in both) or 0.5. If, for example, 1 black cat and two grey cats had been seen, then P[(primaryObjectColor <*GenEnt*> Black)] = 1/3. Facts whose probabilities drop too low are pruned, for efficiency. Importantly, these generalizations do not have logical variables: When non-identical entities are aligned, as in the cats example, a new arbitrary individual (called <*GenEnt*> above) is constructed to stand for the aligned individuals, with its characteristics being determined by the set of statements in the generalization that constrain it.

Extension to Hierarchical Concepts

Our goal is to explore how to extend SAGE to automatically construct psychologically plausible hierarchical concepts. The basic approach is agglomerative hierarchical clustering (Manning et al. 2008) in which the hierarchy is built bottom-up, by merging pairs of existing clusters. The most commonly used hierarchical clustering algorithm is average-linkage clustering, which constructs a dendrogram by merging two most similar members each time, using the mean distance between elements of each cluster as the distance between them. This method is similar to exemplar models, in that similarity is measured by the mean of its members, although computational exemplar models (Medin & Schaffer, 1978; Nosofsky, 1992) use more elaborate combination mechanisms than arithmetic average.

Our approach, starting with SAGE, is more prototype-based. Each SAGE generalization can be thought of as a prototype for the concept represented by that generalization context.

Nearest-Merge Algorithm

Nearest-Merge algorithm uses the same process as average-linkage, but the similarity of two clusters is computed with the generalizations representing each cluster. Each cluster has a *cohesiveness score* which measures the dispersion of exemplars to the generalization. Dispersion denotes how stretched or squeezed a distribution is, calculated here as the average similarity of each exemplar of a generalization to the generalization. The result of merging two clusters depends on their cohesiveness scores relative to the similarity score between the two.

Consider two concepts C1 and C2 with cohesiveness scores c1 and c2, with NSIM(C1,C2) = *s*. If *s* is smaller than c1 and c2, e.g., when we are trying to merge the concepts "blue whale" and "humpback whale", we would construct a superordinate "whale" above them; if s is only larger than one of them, e.g., when we are merging "feline animal" and "cat", the more cohesive category "cat" would break into the other one and becomes a subordinate of it. The situation where s is larger than or equal to c1 and c2 is rare in simultaneous clustering, because usually the more similar pairs would be merged first. But this might be helpful if we have identical examples, for example e1, e2, e3 and e4, they would be merged into one concept (e1, e2, e3, e4) containing all four of them instead of creating a two layer hierarchy of them like ((e1, e2) (e3, e4)). The algorithm is shown in Table 1.

```
procedure nearestMerge (E, a set of one or more examples)
    if numberOf(E) = 1 then:
        return E //a set containing only the root of the
hierarchy, representing the most general concept
    set maxScore = 0, maxPair = nil
    for each example e ∈ E:
        mapping m = macfacBest(e, E - e)
        if NSIM(m) ⩾ maxScore then:
            set maxScore = NSIM(m)
            set maxPair = (m.base,m.target)
    set newConcept =
        conceptMerge(maxPair.base,
                     maxPair.target)
    addSubordinate(maxPair.base, newConcept)
    addSubordinate(maxPair.target, newConcept)
    return
        nearestMerge(E – maxPair + newConcept)

procedure conceptMerge (concept1, concept2)
    set c1 = cohesivenessOf(concept1)
        c2 = cohesivenessOf(concep2)
        ch1= subordinatesOf(concept1)
        ch2= subordinatesOf(concept2)
        score = NSIM( concept1,concept2)
    if score < c1, c2 then:
        newConcept =
            createConcept(concept1 ∪ concept2)
    elseif score ⩾ c1, c2 then:
        newConcept = createConcept(ch1 ∪ ch2)
    elseif c1 > score > c2 then:
        newConcept = createConcept(concept1 ∪ ch2)
    elseif c2 > score > c1 then:
        newConcept = createConcept(ch1 ∪ concept2)
    return newConcept
```

Table 1: Nearest-Merge algorithm

It is useful to be able to flatten a hierarchy into natural clusters, to compare against human clustering results. Note that the cohesiveness score, which estimates the within-category similarity, increases monotonically as we descend to lower level, since lower level concepts are more cohesive. For a given category, let c be its own cohesiveness score, $\bar{c}_{\text{sub}}$ be the average cohesiveness score of its subordinates, and let c_{super} be the cohesiveness score of its superordinate. We cut the hierarchy at the first category along each branch that satisfies $\bar{c}_{\text{sub}} - c < c - c_{\text{super}}$. Intuitively, these natural categories are where the increase of cohesiveness score slows down, which has also been proposed as a criterion for finding the basic level categories (Mervis & Crisafi, 1982)[2]. This algorithm is shown in Table 2. Human categorization is influenced by knowledge and expertise (Murphy, 2004). Experts usually prefer more specific categories, which have higher cohesiveness scores. In contrast, novices usually prefer less specific categories with lower cohesiveness scores. In order to have the flexibility to model both novices and experts, we added an *upper-bound* (0.6 when modelling novices as in Experiment 1 below) and a *lower-bound* (0.8 when modelling experts as in Experiments 3 below), which work as constraints on the cohesiveness score of natural categories (no bounds are used in Experiment 2 below).

procedure ***flattenHierarchy***(*H*, a hierarchy)
    return ***flatten***(***getRoot***(*H*))

procedure ***flatten***(*ct*, a concept)
    if ***leaves?(ct)*** = **true**
      or ***cohesivenessOf***(*ct*) $\geqslant$ *upperBound*
    then:
      return *ct*
    else
      set $\bar{c}_{\text{sub}}$ =
          averageCohesiveness(getSubordinate(ct***))***
        c_{super} =
          cohesivenessOf(getSuperordinate(ct***))***
        c = ***cohesivenessOf(ct)***
      if $(\bar{c}_{\text{sub}} - c < c - c_{\text{super}})$
        and ***cohesivenessOf***(*ct*) $\geqslant$ *lowerBound*
      then:
        return *ct*
      else
        flatCategories = **nil**
        for each *s* ∊ ***getSubordinate(ct)***
          set *flatCategories* =
            flatCategories ∪ ***flatten(s)***
        return *flatCategories*

Table 2: Flattening algorithm

[2] The natural categories we found share some similarity with basic level categories, like being preferred in clustering tasks, but the comparison is not clear because in experiment 1, the input examples are categories instead of individual examples, while in experiment 2 and 3, the basic level categories are not easy to define.

Experiments

We test this clustering algorithm on three datasets. We use average-linkage, with the distance vector computed from structural similarity and content vector dot products (cosine similarity), as baselines for comparison.

Experiment 1: Animal data set

We use the data set of 50 mammals collected by Osherson et al. (1991). Each animal is rated along 85 features, such as having hooves, a long neck, being a quadruped, and so on. These features are converted to binary values using the global mean as the criterion (Kemp et al, 2006). We asked 5 raters to distribute them into natural categories. Although the conditions are different, our result corresponds well with Osherson's. The average number of clusters raters created was 11.4 (SD=3.5), with a minimum of 6 and a maximum of 17, while the average number of clusters in Osherson's data on 30 subjects is 11.5 (SD=3.49) with a minimum of 5 and a maximum of 20.

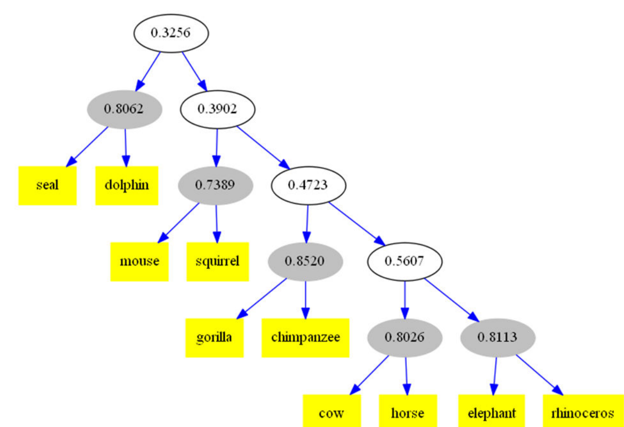

Figure 1: An example of a hierarchy constructed for ten animals, with the cohesiveness scores for the generalizations. The corresponding flat categories are colored in gray.

We used our algorithms to compute clusters for this data set, and then calculated the average ARI (adjusted Rand index) between each algorithm's result and the human-generated clusters. ARI is a commonly used measure of the similarity between two data clusters. It ranges from [-1.0, 1.0], with 1.0 for perfect match, close to 0.0 for random clustering and negative values for bad clusters. We use ARI as a proxy for estimating the psychological plausibility of clusters. We also compare them to the average inter-rater ARI, to provide a benchmark. Table 3 describes the results.

	Inter-rater	Nearest-Merge
Average ARI	0.3939	0.4014
	Average-Linkage(SME)	Average-Linkage(CV)
Average ARI	0.4011	0.4011

Table 3: Results of the animal dataset

The Nearest-Merge algorithm generates 11 clusters, close to the mean of human results (11.4), and average ARI (0.4014) which also corresponds well with inter-rater ARI (0.3939). The average-linkage baseline algorithm requires manual entry of the desired number of clusters, giving it an advantage. The ARI differs little from content vector to SME-based average-linkage (0.4011) and Nearest-Merge, which is to be expected given that the dataset are purely feature-based.

Experiment 2: Political data set

This experiment moves one step into relational structure, using a dataset containing first-order relations. The political dataset, including 14 countries during the cold war, was used by Kemp et al (2006). It included various properties for countries as well as relationships between countries (e.g. that China provides Egypt with economic aid).

Bayesian model	**Nearest-merge**
(Brazil Netherlands) **(UK USA)** **(Burma Indonesia Jordan)** **(India Israel Egypt)** **(Cuba Poland USSR China)**	**(Brazil Netherlands)** **(UK USA)** **(Burma Indonesia) (Jordan)** **(India Israel) (Egypt)** **(Cuba Poland USSR) (China)**

Table 4: Comparison of clusters generated by Bayesian model and Nearest-Merge algorithm

Since no human categorization results are available, we compare our result with the result from Bayesian model (Kemp et al, 2006). The clusters found by the Nearest-Merge algorithm are similar to those found by the Bayesian approach (ARI = 0.6286). Table 4 shows the corresponding clusters. Nearest-Merge tends to treat exceptions like China and Egypt as separate clusters, resulting in finer-grained clusters. The more inclusive clustering by Kemp et al. (2006) might result from their algorithm using a Chinese Restaurant Process, which prefers smaller numbers of clusters.

Experiment 3: Student hand-drawn sketches

Finally, we move to more fully structured data, containing higher-order relations (Table 5). This dataset consists of fault identification worksheets collected by (Chang & Forbus, 2013) with CogSketch (Forbus et al. 2011), an open-domain sketch understanding system. The sketches were created by students taking an undergraduate geoscience course at Northwestern University. There are 28 sketches, drawn from three different exercises. The ground truth clusters were provided by one of the authors of these geoscience exercises (Figure 2 illustrates).

Students are asked to identify and label the fault, hanging wall, foot wall, marker beds, and movement along the fault in the picture by sketching. For example, the sketch in the top left corner of Figure 2 is nearly correct by the exercise's standards. The marker beds are marked by four black rectangles. The two big blue triangle denotes the foot wall (left) and hanging wall (right), and the black arrows show the moving direction of the marker beds.

```
(isa Object-811 GeologicalMarkerBed)
(transMotion Object-811 Down)

(implies
 (and (exceedsQuantInkLeftBound
                      Object-809 Object-446)
      (exceedsQuantInkRightBound
                      Object-809 Object-446))
 (quantInk-tooWide Object-809 Object-446))
```

Table 5: Examples of higher-order relational statements

The sketches are compared with the standard solution provided by the instructor, and turned into relational statements automatically by CogSketch. Table 5 shows some of the statements generated from the bottom left sketch in Figure 2. The statements on the top says that Object-811 is a marker bed and it is moving down. The bottom one says that Object-809, the fault identified by the student, exceeds the left and right bound of Object-446, the corresponding fault in instructor's solution, which implies that Object-809 is too wide compared with Object-446.

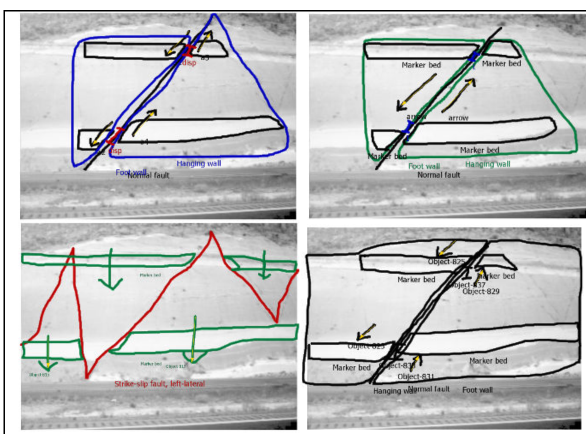

Figure 2: Four sketches from the student sketch dataset. The top two are in the same cluster, while the bottom two each are in their own clusters.

	Average-Linkage (CV)	AL (SME)	Nearest-Merge
Mean ARI, Exercise 1	0.0769	0.4000	0.6842

	AL (CV)	AL (SME)	Nearest-Merge
Mean ARI, Exercise 2	0.4954	0.3836	0.5946

	AL (CV)	AL (SME)	Nearest-Merge
Mean ARI, Exercise 3	0.7037	0.3939	0.7500

Table 6: Sketch dataset results

As Table 6 indicates, the Nearest-Merge algorithm provides the best correspondence with the expert clusters on all three exercises. Notice that there is great variability in how well the SME-only and content-vector scores perform. The difference may depend on whether or not the important properties are as simple as the existence of an entity of a particular type, versus whether or not an important relationship is violated. A content-vector match suffices for detecting whether or not a marker bed is present in a description (leaving out that geological structure turns out to be a common student mistake in the exercises). But detecting that the spatial relationships between two marker beds is incorrect requires a structural match.

General Discussion

To summarize, the Nearest-Merge algorithm results correspond best with human raters, and it produces results comparable with a prior Bayesian model on the dataset for which human data is not available. The performance of average-linkage with content vector and SME score varies on different datasets depending on how much structure information exists and how important this structure is.

Related Work

AI research on conceptual clustering has explored three approaches. The first, and most widely used, approach is to define a distance metric and then search for clusters by optimizing a function of intra-cluster distance and inter-cluster distance (Manning et al. 2008). This approach has been limited mainly to feature vectors, using vector-based distance metric. Our technique can be seen as building on this insight, but by using SME as our model of similarity, we can handle relations and higher-order relations as well as attributes. A second approach is to use Bayesian techniques to produce clusters that maximize predictability and/or utility (Fisher, 1987; Kemp et al 2008). The probability information automatically constructed by SAGE could be used as a source of priors for such algorithms. A third approach is statistical relational learning (Getoor & Taskar, 2007), for example inductive logic programming (Muggleton, 1992) and Markov logic networks (Richardson & Domingos, 2006), which constructs rules via doing statistical reasoning over first-order representations. SAGE generalizations can be used to draw new conclusions via analogical inference, and the probability information it constructs can be used in statistical reasoning to determine when generating rules would be productive (Friedman et al 2009), although that capability is not used here.

Conclusions & Future work

This paper explores how analogical generalization can be extended to model hierarchical concept formation. We show that the Nearest-Merge algorithm can provide psychologically plausible results. Specifically, as the animal data set results indicate, it can produce human-like clusters and hierarchies with the attribute-only representations assumed by research based on feature vectors. As the political data set results indicate, it can create conceptual structures that are compatible with a prior Bayesian simulation, using the same data. As the student sketches data set results indicate, it can produce clusters similar to those generated by an expert, using higher-order relational information. Thus we think this is a promising approach for modeling how people construct hierarchical conceptual structures.

There are several kinds of future work ahead. First, rarely in real life do people have all of the concepts to be organized all at once. Human learning is incremental, and we are experimenting with an incremental version of Nearest-Merge. Robust incremental learning requires overcoming well-known issues with order bias (Eilo & Anderson 1984; Wattenmaker 1993; Kuehne et al 2000). Second, we are exploring better ways to quantitatively measure the similarities and differences in the hierarchies created by people and our models. Classic statistical methods for comparing two hierarchical results (Fowlkes & Mallows, 1983) need the value of average similarity between members for each cluster, which are hard to elicit from human raters, and edit distance metrics are difficult to calculate and score properly for unordered trees. Third, we plan on exploring how these internal, similarity-based criteria interact with linguistic labels, especially when the linguistic labels occur at multiple levels of abstraction and cut hierarchical boundaries (dogs and cats are mammals, and goldfish and sharks are fish, but dogs, cats and goldfish are all pets).

Acknowledgments

This research was sponsored by the Socio-Cognitive Architectures for Adaptable Autonomous Systems Program of the Office of Naval Research, N00014-13-1-0470.

References

Chang, M. D., & Forbus, K. D. (2013). *Clustering Hand-Drawn Sketches via Analogical Generalization.* Twenty-Fifth IAAI Conference.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior, 8*(2), 240-247.

Elio, R., & Anderson, J. R. (1981). The effects of category generalizations and instance similarity on schema abstraction. *Journal of Experimental Psychology: Human Learning and Memory*, *7*(6), 397.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial intelligence, 41*(1), 1-63.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine learning, 2*(2), 139-172.

Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science, 19*(2), 141-205.

Forbus, K. D., Usher, J., Lovett, A., Lockwood, K. and Wetzel, J., 2011. CogSketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science*, 3, 648-666. 43.

Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, *78*(383), 553-569.

Friedman, S., Taylor, J., & Forbus, K. (2009). *Learning naïve physics models by analogical generalization.* Proceedings of the 2nd international analogy conference.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*(2), 155-170.

Gentner, D., & Kurtz, K. J. (2005). Relational categories. *Categorization inside and outside the lab*, 151-175.

Getoor, L., & Taskar, B. (2007). *Introduction to statistical relational learning*: The MIT press.

Hampton, J. A. (1998). Similarity-based categorization and fuzziness of natural categories. *Cognition, 65*(2), 137-165.

Kemp, C., & Tenenbaum, J. B. (2008). The discovery of structural form. *Proceedings of the National Academy of Sciences, 105*(31), 10687-10692.

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., & Ueda, N. (2006). *Learning systems of concepts with an infinite relational model.* Proceedings of AAAI.

Kuehne, S., Forbus, K., Gentner, D., & Quinn, B. (2000). *SEQL: Category learning as progressive abstraction using structure mapping.* Proceedings of the 22nd Annual Meeting of the Cognitive Science Society.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (Vol. 1): Cambridge University Press Cambridge.

McLure, M., Friedman, S., & Forbus, K. (2010). *Learning concepts from sketches via analogical generalization and near-misses.* Proceedings of the 32nd Annual Conference of the Cognitive Science Society (CogSci), Portland, OR.

Medin, D. L. (1983). Structural principles of categorization. In T. Tighe & B. Shepp (Eds.), *Development: Interactional analyses*. Hillsdale , NJ : Erlbaum.

Medin, D. L., Goldstone, R. L., & Gentner, D. (1993). Respects for similarity. *Psychological review, 100*(2), 254.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological review, 85*(3), 207.

Mervis, C. B., & Crisafi, M. A. (1982). Order of acquisition of subordinate-, basic-, and superordinate-level categories. *Child Development*, 258-266.

Mervis, C. B., & Rosch, E. (1981). Categorization of natural objects. *Annual review of psychology, 32*(1), 89-115.

Muggleton, S. (1992). *Inductive logic programming* (Vol. 38): Morgan Kaufmann.

Murphy, G. L. (2004). *The big book of concepts*: MIT press.

Murphy, G. L., & Lassaline, M. E. (1997). Hierarchical structure in concepts and the basic level of categorization.

Nosofsky, R. M. (1992). Similarity scaling and cognitive process models. *Annual review of psychology, 43*(1), 25-53.

Osherson, D. N., Stern, J., Wilkie, O., Stob, M., & Smith, E. E. (1991). Default probability. *Cognitive Science, 15*(2), 251-269.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine learning, 62*(1-2), 107-136.

Rips, L. J. (1989). Similarity, typicality, and categorization. Similarity and analogical reasoning, 21-59.

Tanaka, J. W., & Taylor, M. (1991). Object categories and expertise: Is the basic level in the eye of the beholder?. Cognitive psychology, 23(3), 457-482.

Wattenmaker, W. D. (1993). Incidental concept learning, feature frequency, and correlated properties. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *19*(1), 203.