A Generative Theory of Similarity

Charles Kemp, Aaron Bernstein & Joshua B. Tenenbaum
{ckemp, aaronber, jbt}@mit.edu
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology

Abstract

We propose that similarity judgments are inferences about generative processes, and that two objects appear similar when they are likely to have been generated by the same process. We present a formal model based on this idea, and suggest that it may be particularly useful for explaining high-level judgments of similarity. We compare our model to featural and transformational accounts, and describe an experiment where it outperforms a transformational model.

Keywords: similarity; generative processes; computational theory

Every object is the outcome of a generative process. An animal grows from a fertilized egg into an adult, a city develops from a settlement into a metropolis, and an artifact is assembled from a pile of raw materials according to the plan of its designer. Observations like these motivate the *generative approach*, which proposes that an object may be understood by thinking about the process that generated it. The promise of the approach is that apparently complex objects may be produced by simple processes, an insight that has proved productive across disciplines including biology (Thompson, 1961), physics (Wolfram, 2002), and architecture (Alexander, 1979). To give two celebrated examples from biology, the shape of a pinecone and the markings on a cheetah's tail can be generated by remarkably simple processes of growth. These patterns can be characterized much more compactly by describing their causal history than by attempting to describe them directly.

Leyton has argued that the generative approach provides a general framework for understanding cognition. Applications of the approach can be found in generative theories of memory (Leyton, 1992), categorization (Anderson, 1991; Feldman, 1997; Rehder, 2003), visual perception (Leyton, 1992), speech perception (Liberman et al., 1967), syntax (Chomsky, 1965)[1], and music (Lehrdahl and Jackendoff, 1996). This paper offers a generative theory of similarity, a notion often invoked by models of high-level cognition. We argue that two objects are similar to the extent that they seem to have been generated by the same underlying process.

The literature on similarity covers settings that extend from the comparison of simple stimuli like tones and colored patches to the comparison of highly-structured objects like narratives. The generative approach is relevant to the entire spectrum of applications, but we are particularly interested in high-level similarity. In particular, we are interested in how similarity judgments draw on *intuitive theories*, or systems of rich conceptual knowledge (Murphy and Medin, 1985). Intuitive theories and generative processes are intimately linked: Murphy (1993), for example, defines a theory as "a set of causal relations that collectively generate or explain the phenomena in a domain." Our generative framework should therefore help to explain how similarity judgments are guided by intuitive theories. Others have recognized the importance of this issue: Murphy and Medin (1985) suggest, for example, that "the notion of similarity must be extended to include theoretical knowledge."

We develop a formal theory of similarity and compare it to two existing theories. The featural account (Tversky, 1977) suggests that the similarity of two objects is a function of their common and distinctive features, and the transformation account suggests that similarity depends on the number of operations required to transform one object into the other (Hahn et al., 2003). We show that versions of both approaches emerge as special cases of our model, and present an experiment that directly compares our model with the transformation account.

Generative processes and similarity

Before introducing our formal model, we describe several cases where the assessment of similarity relies on inferences about generative processes. Suppose we are shown

[1] The approach we have described should be distinguished from two usages of "generative" that are found in the linguistics literature. Generativity sometimes refers to the infinite use of finite means: for us, a generative process need not meet this criterion, although many interesting processes will. The second (and more central) usage refers to a grammar's ability to generate the set of grammatical sentences: Chomsky (1965) defines a generative grammar as "a system of rules that in some explicit and well-defined way assigns structural descriptions to sentences." A system of this sort need not be generative in our sense — in particular, it need not assume that structural descriptions have derivational histories. Generative grammars, however, are typically expressed using formalisms that *do* assign derivational histories to structural descriptions, and theories that assume the psychological reality of these histories are instances of the generative approach. Chomsky (1995) has rejected theories of this sort: "the ordering of operations [in grammatical theory] is abstract, expressing postulated properties of the language faculty of the brain, with no temporal interpretation implied." Others, however, argue for linguistic theories that are generative in our sense (Marantz, To appear)

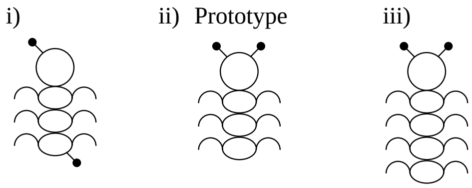

Figure 1: Three bugs. Given that the prototype has been observed, which is more likely to exist — i or iii?

a prototype object and asked to predict what similar objects might exist in the world. There are two kinds of predictions: small perturbations of the prototype, or objects produced by small perturbations of the process that generated the prototype. The second strategy is likely to be more successful than the first, since many perturbations of the prototype will not arise from any plausible generative process, and thus could never appear in practice. By definition, however, an object produced by a small perturbation of an existing generative process will have a plausible causal history.

To give a concrete example, suppose the prototype is a bug generated by a biological process of growth (Figure 1ii). The bug in i is a small perturbation of the prototype, but has no plausible generative history. The bug in iii does have a plausible generative history — it is easy to imagine how a perturbation of the process that produced ii could produce a bug with an extra segment. If we hope to find a bug that is similar but not identical to the prototype, we should look for iii rather than i.

A sceptic might argue that this prediction task can be solved by taking the intersection of the set of objects similar to the prototype and the set of objects that are likely to exist. The two sets could be computed by independent mental modules: the second set depends critically on generative processes, but the first set (and therefore the similarity module) need not. We think it more likely that the notion of similarity is ultimately grounded in the world, and that it evolved for the purpose of comparing real-world objects. If so, then knowledge about what kinds of objects are likely to exist should be deeply bound up with the notion of similarity.

This prediction task is of practical importance, but is not the standard context in which similarity is discussed. More commonly, subjects are shown a pair of objects and asked to rate the similarity of the pair. Note that both objects are observed to exist and the previous argument does not apply. Yet generative processes are still important, since they help pick out the features critical for the similarity comparison. Suppose, for instance, that a forest-dweller discovers a nutritious mushroom. Which is more similar to the mushroom: a mushroom identical except for its size, or a mushroom identical except for its color? Knowing how mushrooms are formed suggests that size is not a key feature. Mushrooms grow from small to large, and the final size of a plant depends on factors like the amount of sunlight it received and the fertility of the soil that it grew in. Reflections like these suggest that the differently-sized mushroom should be judged more similar.

A final reason why generative processes matter is that they are deeply related to essentialism. Medin and Ortony (1989) note that "surface features are frequently constrained by, and sometimes generated by, the deeper, more central parts of objects." Even if we observe only the surface features of two objects, it may make sense to judge their similarity by comparing the deeper properties inferred to generate the surface features. Yet we can say more: just as surface features are generated by the essence of the object, the essence itself has a generative history. Surface features are often reliable guides to the essence of an object, but the object's causal history is a still more reliable indicator, if not a defining criterion of its essence. Keil (1989) discusses the case of an animal that is born a skunk, then undergoes surgery that leaves it looking exactly like a raccoon. Since the animal is generated in the same way as a skunk (born of skunk parents), we conclude that it remains a skunk, no matter how it appears on the surface.

These examples suggest that the generative approach may be broadly useful in explaining high-level similarity judgments. Yet it is unlikely that the approach will be able to account for *all* kinds of similarity judgments. We claim only that there is an important class of judgments that is better explained by the generative approach than by previous approaches to similarity.

We now present a formal model that attempts to capture the intuitions behind the examples described thus far. The rigor of a computational theory is bought at a price, however, and we will only apply the theory to examples much simpler than those already given.

A computational theory of similarity

Given a domain D, we develop a theory that specifies the similarity between any two samples from D. A sample from D will usually contain a single object, but working with similarities between sets of objects is useful for some applications. We formalize a generative process as a probability distribution over D that depends on parameter vector θ.

Suppose that s_1 and s_2 are samples from D. We consider two hypotheses: H_1 holds that s_1 and s_2 are independent samples from a single generative process, and H_2 holds that the samples are generated from two independently chosen processes. Similarity is defined as the log likelihood ratio (Good, 1984), which measures the weight of evidence for H_1 compared to H_2:

$$\begin{aligned}\mathrm{sim}(s_1, s_2) &= \log\left[\frac{P(s_1, s_2|H_1)}{P(s_1, s_2|H_2)}\right] \\ &= \log\left[\frac{\int P(s_1|\theta)P(s_2|\theta)p(\theta)d\theta}{\int P(s_1|\theta)p(\theta)d\theta \int P(s_2|\theta)p(\theta)d\theta}\right] \quad (1)\end{aligned}$$

Equation 1 is not the only way to formalize the generative approach to similarity, and Jebara et al. (2004) describe an alternative that is motivated by similar intuitions. Our model has a clearer probabilistic interpretation than theirs, but the two may well perform similarly in practice.

For some applications, Equation 1 may be difficult to calculate and we will approximate it by replacing the integrals with likelihoods at the maximum *a posteriori* (MAP) values of θ:

$$\mathrm{sim}(s_1, s_2) = \log\left[\frac{P(s_1|\theta_{12})P(s_2|\theta_{12})p(\theta_{12})}{P(s_1|\theta_1)p(\theta_1)P(s_2|\theta_2)p(\theta_2)}\right] \quad (2)$$

where $\theta_{12} = \mathrm{argmax}_\theta P(s_1, s_2|\theta)$, $\theta_1 = \mathrm{argmax}_\theta P(s_1|\theta)$, and $\theta_2 = \mathrm{argmax}_\theta P(s_2|\theta)$.

Similarity is symmetric under this measure: $\mathrm{sim}(s_1, s_2) = \mathrm{sim}(s_2, s_1)$. Whether a symmetric measure is suitable will depend on the context in subtle ways. Consider, for example, the difference between the questions 'How similar are s_1 and s_2?' and 'How similar is s_1 to s_2?' If an asymmetric measure is required, the similarity of s_1 to s_2 could be defined as the probability that s_1 is produced by the process that generated s_2, or that s_2 is produced by the process that generated s_1. This paper, however, will focus on the symmetric case.

We now demonstrate our generative framework in action by deriving a featural model and a transformational model as special cases. [2] Understanding the formal relationships between these models is important for choosing between them, an issue we will soon address.

Featural models

Suppose that objects are represented as binary feature vectors, and let s_1 and s_2 be two objects, $s_1 \cup s_2$ be the set of features shared by both objects, and $s_1 - s_2$ and $s_2 - s_1$ be the sets of features possessed by one object but not the other. Tversky's contrast model proposes that

$$\mathrm{sim}(s_1, s_2) = \gamma_1 F(s_1 \cup s_2) - \gamma_2 F(s_1 - s_2) - \gamma_3 F(s_2 - s_1)$$

where γ_1, γ_2, and γ_3 are positive constants and $F(\cdot)$ measures the saliency of a feature set.

Let n be the number of features possessed by one or both of the objects. To apply our generative framework, let the domain D be the set of all n-place binary vectors. A generative process over D is specified by a n-place vector θ, where θ^i is the probability that an object has value 1 on feature i. We place independent beta priors on each θ^i:

$$\theta^i \sim \mathrm{Beta}(\alpha, \beta)$$
$$s^i \sim \mathrm{Bernoulli}(\theta^i),$$

where s^i is the ith feature value for object s, α and β are hyperparameters and $\mathrm{Beta}(\cdot, \cdot)$ is the beta function.[3] This generative process is known by statisticians as the beta-Bernoulli model, and has previously appeared in the psychological literature as part of Anderson's rational analysis of categorization (Anderson, 1991).

Using Equation 1, we can show that

$$\mathrm{sim}(s_1, s_2) = k_1|s_1 \cup s_2| - k_2|s_1 - s_2| - k_2|s_2 - s_1|$$

where $k_1 = \log\left(\frac{\alpha+1}{\alpha}\right) - \log\left(\frac{\alpha+\beta+1}{\alpha+\beta}\right)$, $k_2 = \log\left(\frac{\alpha+\beta+1}{\alpha+\beta}\right)$, and $F(X) = |X|$ is the cardinality of X.

Under a suitable choice of generative process, then, our model becomes equivalent to a version of the contrast model where $\gamma_2 = \gamma_3$ and $F(\cdot) = |\cdot|$. Our rederivation of Tversky's result makes at least two contributions. First, it provides an interpretation of k_1 and k_2: these parameters are functions of α and β, which make statements about properties of the world. $\frac{\alpha}{\alpha+\beta}$ is the *a priori* probability that an object has any given feature, and $\alpha + \beta$ measures the confidence we should place in this probability. In contrast, the parameters γ_1, γ_2 and γ_3 in Tversky's model are free parameters with no real meaning independent of the model. A second contribution is that our approach automatically provides a setwise similarity measure if s_1 and s_2 are sets of feature vectors rather than single objects. Setwise measures are needed by some psychological models (Osherson et al., 1990), but cannot be derived from the contrast model without additional assumptions.

Transformational models

The transformational approach holds that s_1 is similar to s_2 if s_1 can be readily transformed into s_2. Suppose we are given a set of objects D and a set of transformations T. We assume that every transformation is reversible — if there is a transformation mapping s_1 into s_2, there must also be a transformation mapping s_2 into s_1. A generative process over D is specified by a prototype $\theta \in D$ chosen from a uniform (and possibly improper) distribution over D. To generate an object s from this process, we sample k from a geometric distribution, choose k transformations at random from T, then apply them to the prototype:

$$\theta \sim \mathrm{Uniform(D)}$$
$$k \sim \mathrm{Geometric}(\lambda)$$
$$t^i \sim \mathrm{Uniform}(T)$$
$$s = t^k \cdot t^{k-1} \ldots \cdot t^1(\theta)$$

where λ is a constant, and t^i is the ith transformation chosen. Intuitively, this process tends to generate small variations of the chosen prototype θ, where the permissible variations depend on the set of transformations.

We use Equation 2, and approximate each term in the expression using MAP settings of k and t. The denominator drops out, and the numerator is approximated using

$$\begin{aligned} &P(s_1|\theta_{12})P(s_2|\theta_{12}) \\ &\quad\approx P(s_1|\theta_{12}, \hat{k_1}, \hat{t_1})P(s_2|\theta_{12}, \hat{k_2}, \hat{t_2})P(\hat{k_1}, \hat{k_2}, \hat{t_1}, \hat{t_2}) \\ &\quad= P(\hat{k_1}, \hat{k_2}, \hat{t_1}, \hat{t_2}) \end{aligned}$$

where $\hat{k_1}$ is the number of transformations needed to generate s_1 from the prototype θ_{12}, $\hat{t_1}$ is the set of these transformations, and θ_{12}, $\hat{k_1}$, $\hat{k_2}$, $\hat{t_1}$ and $\hat{t_2}$ are set to values that maximize $P(\theta_{12}, \hat{k_1}, \hat{k_2}, \hat{t_1}, \hat{t_2}|s_1, s_2, H_2)$. Since

[2] Spatial models also emerge as a special case: see the supplementary information at `www.mit.edu/~ckemp/` for details, and for derivations of all results presented here.

[3] If we want to weight features differently, a different α and β can be used for each feature.

there is a cost for each transformation (the geometric distribution encourages $\hat{k_1}$ and $\hat{k_2}$ to be small), the MAP settings for $\hat{k_1}$ and $\hat{k_2}$ minimize the sum $\hat{k_1} + \hat{k_2}$. The minimal value is achieved when $\hat{k_1} + \hat{k_2}$ is the length of the shortest path joining s_1 and s_2, and θ_{12} lies somewhere along this path. It is now straightforward to show that $\text{sim}(s_1, s_2)$ is inversely related to $\hat{k_1} + \hat{k_2}$, or the transformation distance between s_1 and s_2. We suspect that a similar analysis can be given if we relax the assumption that transformations are reversible, although we leave the details for future work.

Choosing between models of similarity

We believe that the featural model, the transformational model and our generative model offer precisely the same expressive power. The featural model can capture an arbitrary dataset perfectly if we have complete freedom to choose the features, and so can the other models if we have complete freedom to choose the transformations or generative processes.[4] It follows that all of the models can mimic each other — given a particular choice of features for the featural model, for example, there will be transformations and generative processes that allow the other models to make exactly the same predictions.

Even though the models have the same expressive power, we can choose between them on grounds of explanatory power. Whenever these models are applied, advocates of each approach need to explain why they chose the features, transformations, and generative processes that they did, and these explanations are unlikely to be equally convincing. Suppose, for example, that the features needed by the featural model seem rather complicated, and share only one property: all of them are signatures of an underlying generative process. Keil's skunk example (Keil, 1989) is one case where this seems to be true, and where the generative approach should probably come out on top. We expect there to be other cases where the featural model is judged superior, and others still where the transformational approach gives the most natural account of the data.

It may be possible to characterize the settings where each of the three models is likely to prove the model of choice. We do not attempt that here, but suggest only that the generative approach is uniquely well-suited to explaining high-level similarity judgments. High-level judgments are likely to rely on intuitive theories, and intuitive theories often specify exactly the kind of information needed by the generative approach: information about the causal histories of objects.

We also believe that there are low-level applications where the generative approach is more explanatory than the other approaches. To support this point and to compare our approach to a published model, we designed an experiment using colored strings as stimuli.

[4]The model specified by Equation 1 can only capture symmetric similarity measures, but as mentioned earlier, there are versions of the generative approach that are not subject to this limitation.

Experiment

Models: We compared the transformational approach to the generative approach in the domain of colored strings. An advantage of choosing this domain is that there are instances of the competing approaches that seem natural but make different predictions. Two indications that the models are natural are that both draw on previously published work, and that neither was developed specifically for this comparison.

The transformation model for binary strings uses the five transformations proposed by Imai (1977) and adopted by Hahn et al. (2003): insertion, deletion, phase shift (shifting all squares one position to the right or left), mirror-imaging (reflection about the central axis), and reversal (the transformation that maps white squares into black squares and vice versa). We extend these transformations to ternary strings in the natural manner. All of the transformations are weighted equally, and the dissimilarity between two strings is defined as the number of transformations required to transform one into the other.

We implement the generative approach using Hidden Markov Models (HMMs), a class of generative processes that is standard in fields including computational biology and computational linguistics. A HMM is determined by a set of internal states, a matrix of transition probabilities q that specifies how to move between the states, and a matrix of observation probabilities o that specifies how to generate symbols from each state. To generate a sequence from a HMM, we choose an initial state from a distribution π, probabilistically generate a color using o, then probabilistically choose the next state using q. We continue until some stopping criterion has been satisfied.

A HMM can be represented using a vector $\theta = \{\pi, o, q\}$. Any given θ induces a probability distribution over the set of all strings, and we can therefore apply the formal model developed above. For simplicity, we use uniform priors on each component of θ and follow the MAP approach in Equation 2. MAP values of θ were computed using the EM algorithm (Murphy, 1998).

Task: We used a forced-choice triad task. Subjects were shown a prototype string, and asked to decide which of two strings was most similar to the prototype. One of these strings was the 'HMM string,' the string most similar to the prototype according to the generative model. The other was the 'transformation string,' the string most similar to the prototype according to the transformational model. Each subject assessed 20 binary triads then 16 ternary triads. Five binary triads are shown in Figure 2, and the full set is available from `www.mit.edu/~ckemp/`.

The triads were chosen systematically to cover most kinds of strings that can be represented using HMMs with a handful of states. We generated a comprehensive set of *HMM types*, then designed a few triads for each type. A HMM type includes an architecture (a graph with arrows indicating probable transitions between states) and a *purity* parameter for each state. A pure state generates only one color, but a noisy state generates multiple colors. Figure 2 shows several of the

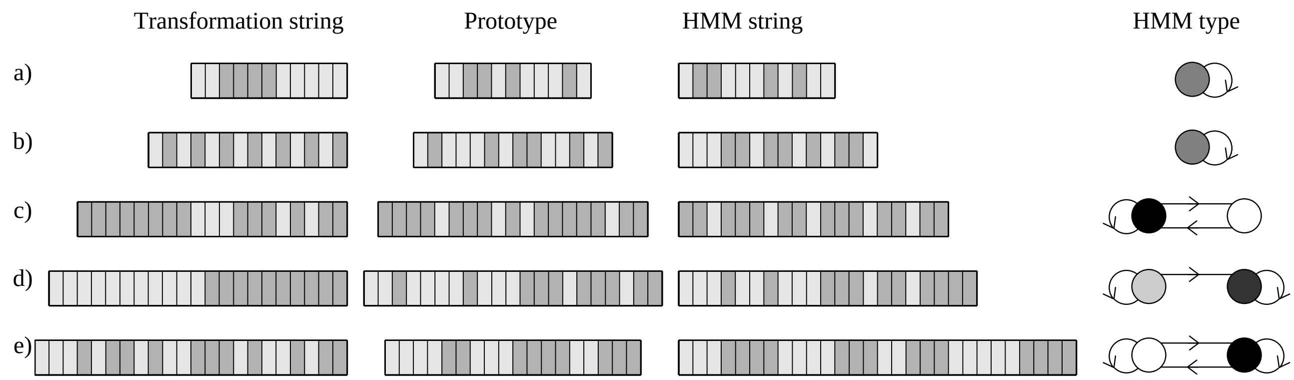

Figure 2: Five binary triads used in the experiment. For each triad, at least 9 of 12 subjects chose the HMM string. The prototype and HMM strings are consistent with the HMM types shown on the right. Arrows indicate high-probability transitions, and the darkness of a state shows its probability of generating the color black.

HMM types used to generate binary strings. The HMM type in 2e moves between a state that generates white squares and another that generates black squares, and tends to generate several squares from each state.

Given a HMM type, we chose a prototype string and a HMM string consistent with the type. The HMM string was usually, but not always the same length as the prototype string. The transformation string was created by transforming the prototype string at a few key points. Two or three transformations were used to create most of the binary transformation strings. The ternary strings are longer, and between three and five transformations were used in most cases.

Results: Table 1 shows results for 12 subjects. There were 240 judgments overall for the binary strings (20 for each subject), and 73% of these judgments favored the generative model. For 17 out of the 20 triads, a majority of subjects chose the generative string, and no no triad clearly favored the transformation model (7 out of 12 subjects chose the transformation string on the most successful triad for this model). The general pattern of results was similar for the ternary strings, but this time a handful of triads clearly favored the transformation model. Overall, these results suggest that similarity judgments between sequences are sensitive to regularities that can be expressed using HMMs.

A possible response is that all of the prototype strings were consistent with simple HMMs, and it is not surprising that a model based on HMMs should perform better than an alternative model. It is true that our sample of strings was biased towards strings generated by simple processes, and is therefore unrepresentative of the set of all possible strings. We suggest, however, that samples from real-world domains are biased in precisely the same way — indeed, that is one of the motivations for our approach. Consider the set of all possible animals, which includes creatures like the manticore, a beast with a man's face, a lion's body and a scorpion's tail. We can imagine animals that are much more bizarre than the manticore, but any sample of real-world animals will be biased towards animals generated by a relatively simple process — descent with modification.

Data	**Judgments**	**Triads**
Binary triads	73	85
Ternary triads	63	69
All triads	69	78

Table 1: Percentages of judgments and of triads that favored the generative model. A triad favored the generative model if more than half of the subjects chose the HMM string.

Discussion

Our results suggest some conclusions about the generative and transformational approaches that apply well beyond the domain of strings. A major problem with the transformational account is that it does not distinguish between generic and non-generic configurations (Jepson and Richards, 1993). Consider the strings in Figure 2a. The transformation string is only two transformations away from the prototype string, but the transformation string is non-generic : since the dark squares appear in a clump, it has a Gestalt property that is not shared by the prototype string. Figure 3a shows another example. The difference between a.i and a.ii is that all the dots have been shifted by a small amount, but a.i is non-generic — it has a striking property that is missing from a.ii.

The generative approach deals neatly with generic and non-generic configurations. The configuration in a.i is most likely to have been generated by a process that produces dots arrayed along a line, and this process has no chance of producing a.ii. The configuration in a.ii is most likely to have been generated by a process that produces a line-shaped cloud of dots, and generating a stimulus like a.i would be an astonishing coincidence under such a process. It follows that a.i and a.ii are unlikely to have been generated by the same process, even though a very small transformation will convert one into the other.

Another way to state the problem is that simple transformations will not suffice for the transformational approach. Consider the stimuli in Figure 3b. Removing an edge between a pair of nodes must be an acceptable transformation, since b.ii is very similar to b.iii, which is identical except for a missing edge. Yet the remove edge

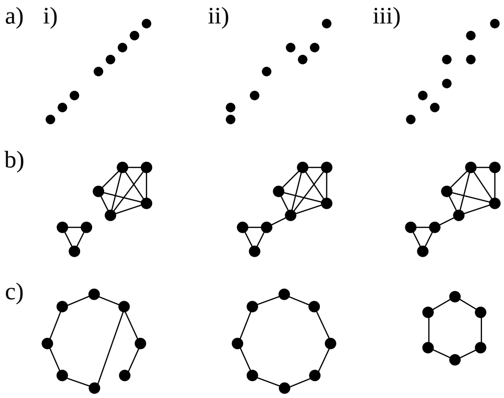

Figure 3: Each central object is a small perturbation of the object on the left, but seems more similar to the object on the right.

transformation must be highly context-sensitive: since b.ii is more similar to b.iii than b.i, it must be more expensive to convert b.ii into b.i. This example suggests that each transformation must be assigned a cost that depends on global properties of the stimulus.

Colored strings are relatively unstructured objects, but we can handle more complex domains using processes that generate structured objects. Kemp et al. (2004), for example, describe a process that generates systems of relations. Analogies form one special family of comparisons between relational systems, and we believe that the generative approach offers a view of analogy that is is intriguingly different from previous approaches. Existing models generally assume that systems are analogous to the extent that there is a structure-preserving one-to-one map between their elements (Gentner, 1983). The generative approach, however, allows analogous systems to have very different numbers of elements, as long as they appear to have been produced by the same process. Consider, for instance, the graphs in Figure 3c. Even though there is a better structure-preserving map between c.ii and c.i, c.ii seems more analogous to c.iii. This is only one suggestive example, but we believe that the generative approach to analogy deserves further investigation.

We have argued that similarity judgments are inferences about generative processes, and suggested how this idea applies to domains ranging from the simple (feature vectors) to the complex (graphs and other structured objects). The generative processes formalized here have been simpler than the processes that appear in people's intuitive theories, but we are optimistic that our framework will help explain how similarity judgments are guided by sophisticated theoretical knowledge.

Acknowledgments We thank Ashish Kapoor for pointing us to (Jebara et al., 2004), and NTT Communication Science Laboratories and DARPA for research support. JBT was supported by the Paul E. Newton Career Development Chair.

References

Alexander, C. (1979). *The timeless way of building.* Oxford University Press, Oxford.

Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review*, 98(3):409–429.

Chomsky, N. (1965). *Aspects of the theory of syntax.* MIT Press, Cambridge, MA.

Chomsky, N. (1995). Categories and transformations. In *The Minimalist Program.* Cambridge.

Feldman, J. (1997). The structure of perceptual categories. *Journal of Mathematical Psychology*, 41:145–170.

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

Good, I. J. (1984). The best explicatum for weight of evidence. *Journal of Statistical Computation and Simulation*, 19:294–299.

Hahn, U., Chater, N., and Richardson, L. B. C. (2003). Similarity as transformation. *Cognition*, 87:1–32.

Imai, S. (1977). Pattern similarity and cognitive transformations. *Acta Psychologica*, 41:433–447.

Jebara, T., Kondor, R., and Howard, A. (2004). Probability product kernels. *Journal of Machine Learning Research*, 5:819–844.

Jepson, A. and Richards, W. (1993). What makes a good feature? In Harris, L. and Jenkin, M., editors, *Spatial vision in humans and robots*, pages 89–121. Cambridge University Press, Cambridge.

Keil, F. C. (1989). *Concepts, kinds, and cognitive development.* MIT Press, Cambridge, MA.

Kemp, C., Griffiths, T. L., and Tenenbaum, J. B. (2004). Discovering latent classes in relational data. AI Memo 2004-019, MIT.

Lehrdahl, F. and Jackendoff, R. (1996). *A generative theory of tonal music.* MIT Press, Cambridge, MA.

Leyton, M. (1992). *Symmetry, causality, mind.* MIT Press, Cambridge, MA.

Liberman, A. M., Cooper, F. S., Shankweiler, D. P., and Studdert-Kennedy, M. (1967). Perception of the speech code. *Psychological Review*, 74:431–361.

Marantz, A. (To appear). Generative linguistics within the cognitive neuroscience of language. *The Linguistic Review.*

Medin, D. and Ortony, A. (1989). Psychological essentialism. In Vosniadou, S. and Ortony, A., editors, *Similarity and analogical reasoning*, pages 179–195. Cambridge University Press, Cambridge.

Murphy, G. L. (1993). Theories and concept formation. In *Categories and concepts: theoretical views and inductive data analysis*, pages 173–200.

Murphy, G. L. and Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological Review*, 92:289–316.

Murphy, K. (1998). Hidden Markov Model (HMM) toolbox for matlab.

Osherson, D. N., Smith, E. E., Wilkie, O., Lopez, A., and Shafir, E. (1990). Category-based induction. *Psychological Review*, 97(2):185–200.

Rehder, B. (2003). A causal-model theory of conceptual representation and categorization. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29:1141–1159.

Thompson, D. W. (1961). *On growth and form.* Cambridge University Press.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84:327–352.

Wolfram, S. (2002). *A new kind of science.* Wolfram media, Champaign, IL.