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 argue 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 describe a formal model based on this idea and show how featural and spatial models emerge as special cases. We compare our approach to the transformational approach, and present an experiment where our model performs better than a transformational model.

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 [18], physics [21], and architecture [1]. 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 perception [12], memory [12], language [3], categorization [2], and music [11]. 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 [15]. Generative processes and theories are intimately linked. Murphy [14], for example, defines a theory as 'a set of causal relations that collectively generate or explain the phenomena in a domain.' We hope that our generative theory provides a framework in which to model how similarity judgments emerge from intuitive theories.

We develop a formal theory of similarity and compare it to three existing theories. The featural account [20] suggests that the similarity of two objects is a function of their common and distinctive features, the spatial account suggests that similarity is inversely proportional to distance in a spatial representation, [19] and the transformation account suggests that similarity depends on the number of operations required to transform one object into the other [6]. We show that versions of each of these approaches emerge as special cases of our generative approach, and present an experiment that directly compares our approach with the transformation account. A fourth theory suggests that similarity relies on a process of analogical mapping [5]. We will not discuss this approach in detail, but finish by suggesting how a generative approach to analogy differs from the standard view.

Generative processes and similarity

Before describing our formal model, we give an informal motivation for a generative approach to similarity. Suppose we are shown a prototype object and asked to describe similar objects we might find in the world. There are two kinds of answers: 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 construction, however, an object produced by a 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 seems unlikely to arise since legs are generated in pairs. A perturbation of the generative process might produce a bug with more segments, such as the bug in iii. If we hope to find a bug that is similar but not identical to the prototype, iii is a better bet than i.

A sceptic might argue that this one-shot learning problem can be solved by taking the intersection of the set of objects similar to the prototype and the set of ob-

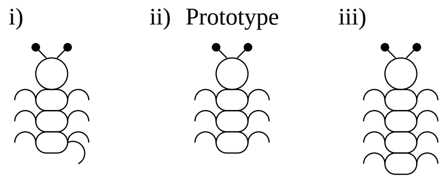

Figure 1: Three bugs. Which is more similar to the prototype — i or iii?

jects that are likely to exist. The second set depends critically on generative processes, but the first set (and therefore the notion of similarity) 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 may be deeply bound up with the notion of similarity.

The one-shot learning problem 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 [13] 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 [9] 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 help to explain a broad class of theory-dependent inferences. We now present a formal model that attempts to capture the intuitions behind all of these cases.

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 probability that the objects are generated by the same process: that is, the relative posterior probability of H_1 compared to H_2:

$$\begin{aligned}\mathrm{sim}(s_1, s_2) &= \frac{P(H_1|s_1, s_2)}{P(H_2|s_1, s_2)} \\ &= \frac{P(s_1, s_2|H_1)P(H_1)}{P(s_1, s_2|H_2)P(H_2)}.\end{aligned}$$

Since we are interested only in the similarity of s_1 and s_2 relative to other pairs of samples, the prior ratio $\frac{P(H_1)}{P(H_2)}$ is a constant and we discard it:

$$\begin{aligned}\mathrm{sim}(s_1, s_2) &\propto \frac{P(s_1, s_2|H_1)}{P(s_1, s_2|H_2)} \\ &= \frac{\int P(s_1|\theta)P(s_2|\theta)p(\theta)d\theta}{\int P(s_1|\theta)p(\theta)d\theta \int P(s_2|\theta)p(\theta)d\theta}.\end{aligned} \tag{1}$$

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:

$$\mathrm{sim}(s_1, s_2) = \frac{P(s_1|\theta_{12})P(s_2|\theta_{12})p(\theta_{12})}{P(s_1|\theta_1)p(\theta_1)P(s_2|\theta_2)p(\theta_2)}, \tag{2}$$

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 may 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 show how to apply our generative framework by deriving a featural model, a spatial model and a transformational model as special cases.

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

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

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:

$$\theta_i \sim \text{Beta}(\alpha, \beta)$$
$$s_i \sim \text{Binomial}(\theta_i),$$

where s_i is the ith feature value for object s, α and β are hyperparameters and $\text{Beta}(\cdot, \cdot)$ is the beta function. This generative process is known by statisticians as the beta-Bernoulli model, and has previously appeared in the psychological literature as part of Anderson [2]'s rational analysis of categorization.

Using Equation 1, we can show that

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

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.[1] Note that $\log(\cdot)$ is a monotonic transformation which can be applied without changing the rank order of the similarities between all pairs of feature vectors.

We therefore see that the generative approach reduces 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. $\frac{\beta}{\alpha+\beta}$ 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 [17], but cannot be derived from the contrast model without additional assumptions.

Spatial models

Spatial models propose that dissimilarity corresponds to the distance between two representations in a multidimensional space. Under a suitable generative process, spatial models also emerge as a special case of our generative framework. Suppose that the domain D is a multidimensional space with dimension n. We formalize a generative process as a Gaussian distribution over D with mean μ and covariance matrix Σ. For simplicity, we place a uniform (hence improper) prior over μ:

$$\mu \sim \text{Uniform}(\mathcal{R}^n)$$
$$s \sim \text{Normal}(\mu, \Sigma),$$

where μ and s are random variables with n dimensions, and Σ is a constant n by n matrix.

Using Equation 1, we can show that

$$\log(\text{sim}(s_1, s_2)) = -(s_1 - s_2)^T \Sigma^{-1} (s_1 - s_2)$$

where again we are interested only in the rank order of the similarities. Under a Gaussian generative process, then, similarity is inversely related to the Mahalanobis distance between two representations. If the covariance matrix is spherical ($\Sigma = \sigma^2 I$, where I is the n-dimensional identity matrix and σ is a constant), then similarity is inversely related to the Euclidean distance between representations:

$$\log(\text{sim}(s_1, s_2)) = -(s_1 - s_2)^T (s_1 - s_2).$$

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 a process, we sample a transformation count k from an exponential distribution, choose k transformations at random from T, then apply them to the prototype:

$$\theta \sim \text{Uniform(D)}$$
$$k \sim \text{Exponential}(\lambda)$$
$$t_i \sim \text{Uniform}(T)$$
$$s = t_k \cdot t_{k-1} \ldots \cdot t_1(d)$$

where λ is a constant, and t_i is the ith transformation chosen. A generative process in this family will tend to produce small variations of the chosen prototype.

We use Equation 2, and approximate each term in the expression using MAP settings of k and t:

$$P(s_1|\theta_1) = \int P(s_1|\theta_1, k, t) P(k) P(t) dk dt \approx P(s|\theta_1, \hat{k}, \hat{t})$$

where θ_1, $\hat{k}$ and $\hat{t}$ are set to values that maximize $P(\theta, k, t|s)$. Since $\theta_1 = s_1$ and $\hat{k} = 0$, $P(s_1|\theta_1, \hat{k}, \hat{t}) = 1$. Similarly, we use

$$P(s_1|\theta_{12}) P(s_2|\theta_{12}) \approx P(s_1|\theta_{12}, \hat{k_1}, \hat{t_1}) P(s_2|\theta_{12}, \hat{k_2}, \hat{t_2}).$$

[1] Full derivations of all results can be found at `www.mit.edu/~ckemp/`

In this case, $\hat{k_1} + \hat{k_2}$ is the length of the shortest path joining s_1 and s_2 where each step along the path is a transformation from T. Since the transformations are reversible, $P(\theta|s_1, s_2)$ is the same for any θ along this path, and we can set θ_{12} to any of these values. It is now straightforward to show that $\log(\text{sim}(s_1, s_2))$ is approximated by the transformation distance between s_1 and s_2, or the length of the shortest path joining these objects. 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.

This derivation does not imply that the generative approach is strictly more powerful than the transformational approach. One can also argue that the generative approach is a special case of the transformational approach. Hahn et al. [6] are careful to note that the similarity of two objects depends on the transformation distance between their *representations*, not between the objects themselves. If each object is represented as a generative process — for example, as the generative process most likely to have created the object — then the transformational account may end up looking similar to the generative approach.

The ability of these approaches to mimic each other is a mark of their expressive power. Expressive power is both a boon and a shortcoming. Classic approaches like the featural and spatial approaches operate over simple representations (feature vectors and multidimensional spaces), and are too limited to capture similarity judgments between complex structured representations. Both the transformational and the generative approaches are powerful enough to deal with complex representations, and we describe later how the generative approach can compute the similarity between systems of relations.

Expressive power is a shortcoming when a powerful approach is able to model any conceivable effect, regardless of whether it matches human data or not. To avoid this problem we can limit the generative approach and the transformational approach by allowing only generative processes, representations and transformations that seem natural. It is difficult to specify precisely what makes a generative process natural, but a natural process should be simple, and motivated if possible by knowledge about how real world objects are generated. A process is unlikely to be natural if it is tendentious, or contrived in order to produce a highly specific result. Of the generative processes described so far, the beta-Bernoulli and Gaussian models are natural, and are widely used in statistics and machine learning. The prototype model seems less natural.

Even though the transformational and the generative approaches may reduce to one another, it does not follow that the two are interchangeable or equally successful. A result that is naturally captured by one may not be naturally captured by the other. To compare the two, we must decide which approach offers greater scope for providing natural explanations of the phenomena we care about.

Experiment

We compared the generative approach with the transformational approach using colored strings as stimuli. An advantage of choosing this domain is that we can formulate instances of the competing approaches that seem natural but make different predictions. An indication that both models are natural is that both draw on previously published work, and neither was developed specifically for this comparison.

The transformation model uses the set of transformations suggested by Imai [7] and adopted by Hahn et al. [6]. These authors suggest five transformations over binary strings: 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 a given 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 generative approach to similarity developed above. For simplicity, we use uniform priors on each component of θ and follow the MAP approach in Equation 2. To compute MAP values of θ we used Murphy [16]'s implementation of the standard EM algorithm for inference in HMMs.

Each experimental 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/`. One triad was presented per screen, and subjects decided whether the leftmost or the rightmost string was most similar to the prototype string. One of these strings was the 'HMM string,' the string most similar to the prototype string according to our generative approach. The remaining string was the 'transformation string' (each triad was chosen so that the two models made different predictions). The left-right order of the HMM and transformation strings was chosen randomly for each screen.

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

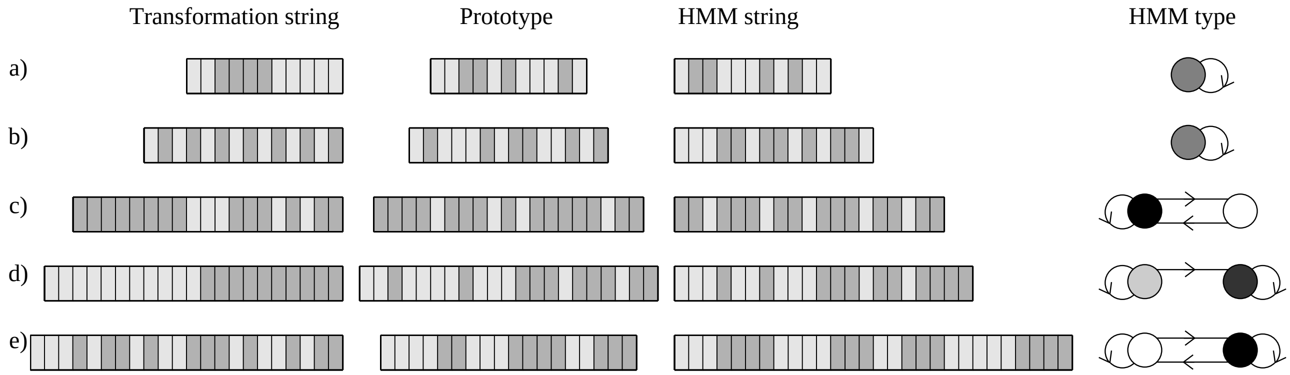

Figure 2: Five binary triads used in the experiment. For each triad, at least 9 of 12 subjects chose the HMM string. The prototype and HMM strings are consistent with the HMM types shown on the right. Arrows indicate high-probability transitions, and the darkness of a state shows its probability of generating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

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 HMM types used to generate binary strings.

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 for 12 subjects are shown in Table 1. For the binary strings, 73% of the judgments favored the generative approach, and a majority of subjects chose the generative string on 17 out of 20 triads. No triad clearly favored the transformation model: 5 out of 12 subjects chose the HMM 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 natural 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.

A second response is that the transformation model performs poorly because we have left out several crucial transformations. The ideal transformation model would include all relevant transformations, just as the ideal generative model would include all relevant generative processes. There may be an additional set of simple transformations that would reverse our findings, but we have been unable to think of it.

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 [8]. 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 i and ii is that all the dots have been shifted by a small amount, but i is non-generic – it has a striking property that is missing from 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 ii is very similar to iii, which is identical except for a missing edge. Yet the remove edge transformation must be highly context-sensitive: in particular, it must be more expensive to convert ii into i than into iii. This example suggests that a given transformation must be assigned a cost that depends on global properties of the stimulus.

Given an appropriate set of high-level features, a featural approach could probably account for our data. In Figure 2e, for example, the prototype and the transformation strings share the property of 'streakiness' and the transformation string alone looks 'irregular.' It is not sufficient, however, to choose a set of features that fits the data. The choice of features must be justified — since the space of possible features is vast, a convincing featural model must use features that seem *natural*. It may be possible to reconcile the featural and the generative approaches by arguing that natural features are those that provide signatures of an underlying generative process.

Colored strings are relatively unstructured objects, but we can handle more complex domains using processes that generate structured objects. Kemp et al. [10], 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 [4]. 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 2c. Even though there is a better mapping between ii and i, ii seems more analogous to 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 comparisons between highly 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 is guided by sophisticated theoretical knowledge.

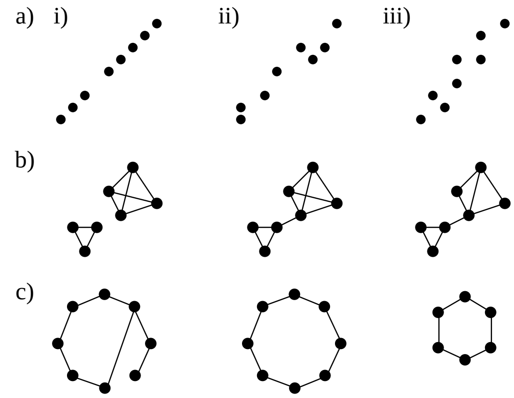

Figure 3: Each central object is a small perturbation of the object on the left, but seems more similar to the object on the right.

References

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

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

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

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

[5] Gentner, D. and Markman, A. B. (1997). Structure mapping in analogy and similarity. *Journal of Mathematical Psychology*, 52:45–56.

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

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

[8] 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.

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

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

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

[12] Leyton, M. (1992). *Symmetry, Causality, Mind.* MIT Press, Cambridge, MA.

[13] 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.

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

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

[16] Murphy, K. (1998). Hidden Markov Model (HMM) toolbox.

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

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

[19] Torgeson, W. S. (1965). Multidimensional scaling of similarity. *Psychometrika*, 30:379–393.

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

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