Supporting Information

Kemp and Tenenbaum 10.1073/pnas.0802631105

We describe our hypothesis space of structural forms in more detail, and formally specify the distributions $P(S|F)$ and $P(D|S)$. We then describe our implementation of our model, and introduce the data sets that led to the results in Figs. 3 and 4. We expand on the developmental shift described in the main text (Fig. 5), and finish by comparing our approach to previous models of structure learning.

All data sets along with code for running our model can be downloaded from `http://charleskemp.com`

A Hypothesis Space of Structural Forms

The first six forms in Fig. 2*A* are primitive forms, each of which can be generated using a node-replacement graph grammar with a single production. To grow a graph, we start with a seed graph and repeatedly split nodes according to the grammar. For all primitive forms except the ring, the seed is a graph with one node and no edges. For the ring, the seed is a single-node graph with a self link.

The remaining forms in Fig. 2*A*—the grid and the cylinder—can be expressed as products of primitive forms. A grid is the Cartesian graph product of two chains, and a cylinder is the product of a ring and a chain.[1] We grow grids by representing the two dimensions separately, and using the chain grammar to grow one of the dimensions. Cylinders can be generated similarly.

When working with feature or similarity data, our hypothesis space of structural forms includes undirected versions of the eight forms in Fig. 2*A*. For example, the undirected version of an order is a fully connected graph. When working with relational data, for convenience we restrict the analysis to graphs where each node represents a non-empty cluster of entities. Trees, grids and cylinders allow nodes to be empty, and we remove these from our collection of structural forms, leaving five forms in total. Given a relation it is important to discover whether the relation tends to hold between elements in the same cluster, and whether the relation is directed or not. The forms in Fig. 2*A* use nodes without self-links, and therefore assume that the relation does not hold within clusters. We create a set of 10 forms by supplementing each form with an alternative that uses nodes with self-links, but is otherwise identical. Each of these 10 forms uses directed edges, and for each we include an additional form with undirected edges. In total, then, our hypothesis space of relational forms includes 20 candidates.[2] The four chain-structured forms in this hypothesis space are shown in Fig. S1.

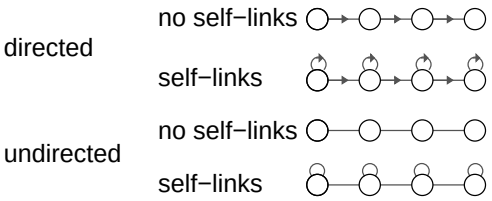

Fig. S1. The four chain-structured forms used for relational data.

A Meta-Grammar for Generating Structural Forms

Although we focus on the eight forms in Fig. 2, it is natural to consider other possibilities. We have suggested that graph grammars provide a unifying language for expressing many different structural forms, and ultimately it may be possible to develop a 'Universal Structure Grammar' that generates all and only the cognitively natural forms.

As an initial step towards this goal, note that all of the grammars in Fig. 2 can be generated from the template in Fig.

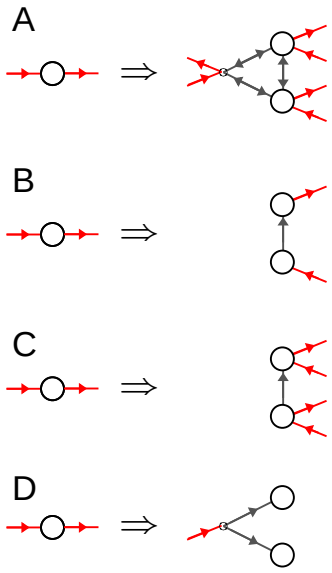

Fig. S2. Generating graph grammars from a meta-grammar. (*A*) The six grammars in Fig. 2*A* correspond to subsets of the template shown here. (*B*,*C*,*D*) Subsets of the production in *A* that grow chains, orders, and trees.

S2*A*. The right-hand side of this template includes 12 arrows, and we can create a range of new productions by removing some of these arrows. Figs. S2 *B*-*D* show how three of the grammars in Fig. 2 correspond to subsets of the template. Combining the template in Fig. S2*A* with a procedure for removing arrows creates a meta-grammar [1] that generates grammars for many structural forms. Some of these forms,

[1] A two dimensional Euclidean space can be generated as the regular Cartesian product of two chains, where each chain is viewed as a continuous object rather than a graph. Our generative model for feature data extends naturally to continuous spaces, but we restrict ourselves here to graph structures.

[2] Only 17 of these forms are actually distinct. A partition (with or without self-links) remains the same when converted to an undirected graph. An undirected order with self links is a fully connected graph, and is very similar to a partition graph without self links (a graph with no edges). In both cases, all clusters stand in the same relationship to each other.

[3] There are methods for learning partitions [2] and trees [3] when the set of entities is countably infinite, and future work should consider whether these methods can be used to develop a framework for learning many kinds of forms.

[4] In the case of trees, internal nodes are required to be empty, but we do not allow empty leaf nodes.

[5] If S is a tree, since entities may only appear at its leaves, we adopt the convention that $|S|$ is equal to the number of leaf nodes in S.

Table. S1. Number of k-cluster structures for several different forms

Form F	C(F, k)
Partition	1
Directed Chain	$k!$
Undirected Chain	$\frac{k!}{2}$
Order	$k!$
Connected	1
Directed Ring	$(k-1)!$
Undirected Ring	$\frac{(k-1)!}{2}$
Directed Hierarchy	$k^{\frac{k-1}{2}}$
Undirected Hierarchy	k^{k-2}
Tree	$(2k-5)!!$

although certainly not all, are likely to be useful for structure discovery. In principle, a learning system could begin with just this meta-grammar and go on to discover any form that is consistent with the meta-grammar.

All of the grammars generated by the meta-grammar in Fig. S2 include just one production, but additional forms can be generated if we allow grammars with multiple productions, and productions where the edges on the right hand side are chosen probabilistically. Our work so far has focused on simple grammars that generate some of the most frequently used forms, but further exploration of the space of grammars is an important direction for future work.

Generating Structures from Structural Forms

Suppose that we are working with n entities.[3] A structure S is a graph where the nodes correspond to clusters of entities. S is compatible with F if S can be generated by the generative process defined for F, and if S contains no empty nodes when projected along any of its component dimensions (Fig. S3).[4] There is a finite collection of structures that are compatible with a given form F, and $P(S|F)$ is non-zero only for graphs in this collection. To encourage the model to choose the simplest adequate representation for a domain, we weight each structure according to the number of nodes it contains:

$$P(S|F) \propto \begin{cases} 0 & \text{if } S \text{ is incompatible with } F \\ \theta^{|S|} & \text{otherwise,} \end{cases} \qquad [\mathbf{S1}]$$

where $0 < \theta \leq 1$, and $|S|$ is the number of nodes in S.[5] For all analyses reported in this paper we set $\theta = e^{-3}$, which means that each additional node reduces the log probability of a structure by 3. In most cases, similar results are found by setting $\theta = 1$, which produces a uniform distribution over structures of a given form. Analyses of synthetic data, however, suggest that a complexity penalty is useful when fitting grids and cylinders. Without this penalty, the model may introduce additional nodes that improve the fit slightly but that do not capture important structural distinctions (Fig. S4).

The normalizing constant for the distribution in Equation S1 is the sum

$$\sum_S P(S|F) = \sum_{S \text{ is compatible with } F} \theta^{|S|}.$$

To compute this quantity, we must consider all possible ways of putting n entities onto a graph of form F. Let $S(n,k)$ be the Stirling number of the second kind: the number of ways

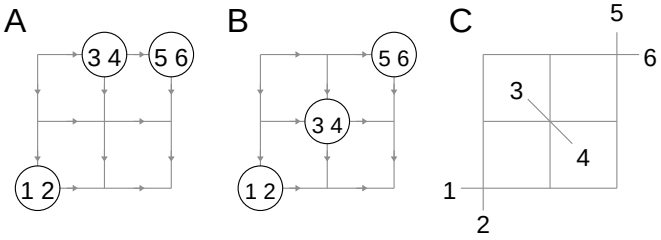

Fig. S3. Cluster graphs and entity graphs. (A) A cluster graph that is incompatible with the grid form, since the middle node will be empty if the graph is projected onto the vertical axis. (B) A cluster graph that is compatible with the grid form. (C) An entity graph corresponding to the cluster graph in (B).

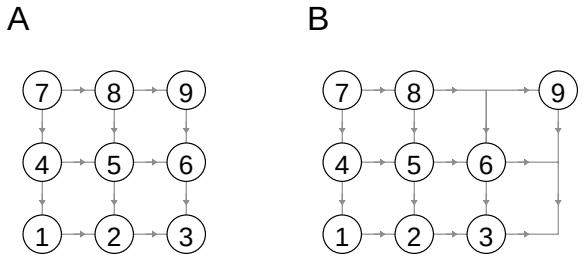

Fig. S4. Capturing a preference for simple structures. (A) Setting $\theta < 1$ encourages the model to find structures with few nodes. The model therefore prefers grids and cylinders where most of the nodes are occupied. (B) Setting $\theta = 1$ produces a uniform distribution over all graphs compatible with a given form. The model may now introduce additional nodes that improve the fit slightly by capturing metric properties (perhaps entities 9 and 8 are less similar than entities 6 and 5), but that do not capture important structural differences.

to partition n elements into k nonempty sets. Let $C(F,k)$ be the number of F-structures with k occupied cluster nodes. Expressions for $C(F,k)$ for all forms except the grid and the cylinder are shown in Table S1. The number of n-entity structures with form F is

$$\sum_{k=1}^{n} S(n,k)C(F,k).$$

For all forms F except the grid and the cylinder, the normalizing constant for Equation S1 is

$$\sum_{S \text{ is compatible with } F} \theta^{|S|} = \sum_{k=1}^{n} S(n,k)C(F,k)\theta^k.$$

This equation groups the F-compatible structures into classes that share the same partition of the entities. To compute the normalizing constant for product structures like the grid and the cylinder, it is more convenient to group the F-compatible structures into classes that share the same basic topology. Let $G(n,i,j)$ be the number of ways to put n entities on an undirected i by j grid so that no dimension of the grid remains unoccupied. The normalizing constant for grids is now

$$\sum_{i \leq j \leq n} G(n,i,j)\theta^{ij}.$$

[6] See [5, 6, 7] for related work on Gaussian graphical models.

Similarly, if $Y(n,i,j)$ is the number of ways to put n entities on an undirected i by j cylinder so that no dimension remains unoccupied, the normalizing constant for cylinders is

$$\sum_{i\leq n, j\leq n} Y(n,i,j)\theta^{ij}.$$

$G(\cdot,\cdot,\cdot)$ can be computed using the function $L(\cdot,\cdot)$, where $L(n,i)$ is the number of ways to put n entities on an undirected i node chain so that no node remains empty:

$$L(n,i) = \begin{cases} 1 & \text{if } i = 1 \\ \frac{i!}{2} S(n,i) & \text{if } i > 1 \end{cases}$$

where $S(n,i)$ is the Stirling number of the second kind.

We now have

$$G(n,i,j) = \begin{cases} L(n,i)L(n,j) & \text{if } i \neq j \\ \frac{L(n,i)^2 + L(n,i)}{2} & \text{if } i = j. \end{cases}$$

In the case where $i = j$, we have accounted for the fact that the grid can be rotated without changing the configuration.

The counts for undirected cylinders can be computed similarly. Define

$$R(n,i) = \frac{L(n,i)}{i}$$

where $R(n,i)$ is the number of ways to put n entities on an i node ring so that no node remains empty. Then

$$Y(n,i,j) = L(n,i)R(n,j).$$

Generating Data from Structures

Suppose that S is a directed graph with nodes that correspond to clusters of entities.

Feature Data

Let D be an entity-feature matrix where the (i,j) entry indicates the value of entity i on feature j. We represent the structure of the set of entities using undirected *entity graphs*. Cluster graphs are converted to entity graphs by adding a node for each entity, connecting each entity to the cluster node that contains it, and replacing each directed edge with an undirected link (Fig. S3). We set $P(D|S) = P(D|S_{ent})$ where S_{ent} is the entity graph corresponding to cluster graph S.

Given an entity graph S_{ent}, we expect nearby entities in the graph to have similar features, and formalize this intuition by assuming that the features are independently generated from a Gaussian distribution over the graph [4].[6] Suppose that S_{ent} is a graph with $n + l$ nodes, where the first n nodes correspond to entities and the remaining l nodes are latent. Let f be a feature vector which assigns a continuous value $f_i \in \mathbb{R}$ to each node i in the graph.

Let W be a $n+l$ by $n+l$ weight matrix, where $w_{ij} = \frac{1}{e_{ij}}$ if nodes i and j are joined by an edge of length e_{ij} and $w_{ij} = 0$ otherwise. We now define the graph Laplacian $\Delta = E - W$ where E is a diagonal matrix with entries $e_i = \sum_j w_{ij}$. A generative model for f that favors features which are smooth over the graph S_{ent} is given by

$$P(f|W) \propto \exp\left(-\frac{1}{4}\sum_{i,j} w_{ij}(f_i - f_j)^2\right) = \exp\left(-\frac{1}{2} f^{\mathsf{T}} \Delta f\right). \quad \textbf{[S2]}$$

Equation S2 indicates that our prior $p(f|W)$ penalizes a feature vector f whenever $f_i \neq f_j$ and i and j are adjacent in the graph, and that the penalty increases as the edge between i and j becomes shorter (i.e. w_{ij} increases).

Zhu et al. [4] point out that Equation S2 can be viewed as a Gaussian prior over f with zero mean and covariance matrix Δ^{-1}. The prior, however, is improper. Note that any feature vector f has the same probability when shifted by a constant, which effectively means that the variance of each f_i is infinite. We obtain a proper prior by assuming that the feature value f_i at any entity node has an *a priori* variance of σ^2:

$$f \,|\, W \sim \mathcal{N}(0, \tilde{\Delta}^{-1}) \quad \textbf{[S3]}$$

where $\tilde{\Delta} = \Delta + V$, and V is a diagonal matrix with $\frac{1}{\sigma^2}$ appearing in the first n positions along the diagonal and 0 elsewhere.[7]

Equation S3 specifies how to generate a single feature only. Typically the data D will include multiple features, and we assume that the features are conditionally independent given S_{ent}.[8] To complete the generative model we place priors on the branch lengths e_{ij} and the variance σ^2. Both are drawn from exponential distributions with hyperparameter β:

$$\begin{aligned} \sigma \,|\, \beta &\sim \text{Exponential}(\beta) \\ e_{ij} \,|\, S_{ent}, \beta &\sim \text{Exponential}(\beta) \text{ if } s_{ij} = 1. \end{aligned}$$

For all analyses we set $\beta = 0.4$.

Even though we have introduced edge weights w_{ij}, we are interested primarily in the best graph topology for the data D. The likelihood $P(D|S_{ent})$ can be computed by integrating out σ and the edge weights:

$$P(D|S_{ent}) = \int P(D|S_{ent}, W, \sigma^2) P(W|S_{ent}) P(\sigma^2) dW d\sigma^2.$$

We approximate this integral using the Laplace approximation. Since the weights w_{ij} and the variance σ are both required to be positive, we transform them to a log scale before computing the Laplace approximation. To find modal values of the transformed variables, we ran a gradient-based search using the 'Large Scale' option available as part of MATLAB's unconstrained minimization routine.

Our generative model for features assumes that the data are continuous, but Figs. 3*A* and 3*B* were learned from binary features. When working with binary data, we treat feature values 0 and 1 as real numbers, and scale the data matrix D as described below so that the mean entry in the matrix is 0. Generative models analogous to Equation S2 can be defined for binary features [8], but structure learning becomes more difficult: in particular, computing $P(D|S)$ is challenging when S is multiply connected. Our decision to work with Gaussian models is motivated by computational issues of this sort, but

[7] Zhu et al. [4] use a matrix V that has $\frac{1}{\sigma^2}$ everywhere along the diagonal. We prefer our approach because it allows empty nodes to be added to a weighted graph W without changing the likelihood $P(D|W)$. Suppose that we convert graph W to W' by adding an empty node k to the edge between i and j so that $d_{ij} = d'_{ik} + d'_{kj}$. Under our model, $P(D|W) = P(D|W')$, but this result does not hold for the approach of [4].

[8] We treat all features equally, but it is possible to introduce weights λ^j for each feature. Equation S3 then becomes $P(f^j) \propto \exp\left(-\frac{\lambda^j}{2} f^{\mathsf{T}} \Delta f\right)$, where f^j is the jth feature. Once we place a prior on the feature weights (for example, a prior that encourages most weights to be small), we can simultaneously discover the structure S and the weights for each feature. The weights will measure the extent to which a feature is smooth over S—the features that match the structure best will end up with the highest weights.

extensions of our approach can explore more principled treatments of discrete features.

Throughout this section, we have not been careful to distinguish between probability density functions and probability distributions. Since we defined a generative model for continuous vectors f, $P(f|W)$ should strictly be written as a probability density function $p(f|W)$. In practice, however, f is only observable to some level of accuracy, and we can quantize each feature vector:

$$P(f|W) = \int_{|f-u|<\epsilon} p(u|W)du \qquad \textbf{[S4]}$$

where ϵ is a small constant. Equation S4 can be approximated as

$$P(f|W) \approx p(f|W) \int_{|f-u|<\epsilon} du \propto p(f|W) \qquad \textbf{[S5]}$$

where the constant of proportionality does not depend on the structure or the form under consideration, and can be dropped from our calculations.

Similarity Data

Under our generative model for features, the data matrix D influences the distribution $P(D|S_{ent})$ only through the number of features m and the covariance matrix $\frac{1}{m}DD^\mathsf{T}$:

$$\log(P(D|W,\sigma)) = -\frac{mn}{2}\log(2\pi) - \frac{m}{2}\log|\tilde{\Delta}^{-1}| - \frac{1}{2}\mathrm{tr}(\tilde{\Delta}DD^\mathsf{T})$$

As long as m and the covariance matrix are provided, our approach to structure discovery can be used even if none of the features in D is actually observed. If we assume that a given (symmetric) similarity matrix is a covariance matrix, we can therefore learn structural forms from similarity data. In many cases the similarity matrix will already be positive definite, but if not we make it so by replacing all negative eigenvalues with zeroes.

Although we have loosely described $\frac{1}{m}DD^\mathsf{T}$ as a covariance matrix, it can be characterized more precisely. If the features in D are generated from a Gaussian distribution with *zero* mean and unknown covariance Σ, then $\frac{1}{m}DD^\mathsf{T}$ is the maximum likelihood estimator of Σ. This matrix differs from the "empirical covariance" found in some textbooks, which is the maximum likelihood estimator if the features in D are generated from a Gaussian distribution with *unknown* mean and unknown covariance. The two estimators coincide if each row of D has a mean of zero. When working with feature data, we normalize D so that the mean value across the entire matrix is zero. In this case, the matrix $\frac{1}{m}DD^\mathsf{T}$ and the empirical covariance are likely to be similar but not identical, and deciding to work with one rather than the other should make little difference.

Relational Data

Suppose now that the data specify relationships between entities rather than features of the entities. We define two generative models, one for frequency data and the other for binary relations. Each model takes a single two-place relation as input—for instance, *dominates*$(\cdot,\cdot)$ or *communicates_with*$(\cdot,\cdot)$. Future work can consider cases where multiple relations must be simultaneously analyzed.

Suppose first that D is a square frequency matrix with a count d_{ij} for each pair of entities (i,j). If the entities are people, for example, d_{ij} may indicate the number of times that person i spoke to person j. We define a generative model where $P(D|S)$ is high if the large entries correspond to edges in the cluster graph S.

Formally, let $|a|$ be the number of entities in cluster a. Let C be a matrix of between-cluster counts, where C_{ab} is the total number of counts observed between entities in cluster a and entities in cluster b. Our model assumes that $P(D|S) = P(D|C)P(C|S)$, and that C is generated from a Dirichlet-multinomial model:

$$\begin{aligned} \theta \,|\, S, \beta_0, \beta_1 &\sim \text{Dirichlet}(\alpha) \\ C \,|\, \theta, n_{\text{obs}} &\sim \text{Multinomial}(\theta) \end{aligned}$$

where $\alpha_{ab} = \beta_0|a||b|$ if $S_{ab} = 0$, $\alpha_{ab} = \beta_1|a||b|$ if $S_{ab} = 1$, and n_{obs} is the total number of observations. The pair (β_0, β_1) is drawn from a discrete space: $\beta_0 + \beta_1$ is drawn uniformly from $\{\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8, 16, 32\}$ and $\frac{\beta_0}{\beta_0+\beta_1}$ is drawn uniformly from $\{0.05, 0.15, \ldots, 0.45\}$. A count matrix C is assigned high probability under this model if the large entries in C tend to correspond to edges in the cluster graph S.

As we did for the feature model, we integrate out the parameters:

$$\begin{aligned} P(C|S) &= \int P(C|S,\beta_0,\beta_1)P(\beta_0,\beta_1)d\beta_0 d\beta_1 \\ &= \frac{1}{50}\sum_{(\beta_0,\beta_1)} P(C|S,\beta_0,\beta_1) \end{aligned}$$

where

$$P(C|S,\beta_0,\beta_1) = \int P(C|\theta)p(\theta|S,\beta_0,\beta_1)d\theta$$

can be computed analytically, since the Dirichlet prior on θ is conjugate to the multinomial $P(C|\theta)$.

Given C, we assume that the C_{ab} counts are distributed at random between all pairs (i,j) where i belongs to cluster a and j belongs to cluster b:

$$P(D|C) = \prod_{a,b}\left(\frac{1}{|a||b|}\right)^{C_{ab}}.$$

Binary Relations

A similar approach can be used to analyze binary relations. Suppose that D is a square binary matrix where d_{ij} is 1 if the relation holds between i and j and 0 otherwise. In a social setting, for instance, d_{ij} may indicate whether i gives orders to j. We define a generative model where $P(D|S)$ is high if the non-zero entries in D tend to correspond to edges in the cluster graph S.

Given a cluster graph S, let z_i denote the cluster assignment for entity i. Suppose that there is a parameter θ_{ab} for each pair of clusters, and that d_{ij} is generated by tossing a coin with bias $\theta_{z_i z_j}$. We place a prior distribution on the parameters θ_{ab} that depends on the edges in the cluster graph, and that encourages d_{ij} to be true when there is an edge between cluster z_i and cluster z_j. The model can be written as:

$$\begin{aligned} \theta_{ab} \,|\, S, \alpha_0, \beta_0, \alpha_1, \beta_1 &\sim \begin{cases} \text{Beta}(\alpha_0, \beta_0), & \text{if } S_{ab} = 0 \\ \text{Beta}(\alpha_1, \beta_1), & \text{if } S_{ab} = 1 \end{cases} \\ d_{ij} \,|\, \theta &\sim \text{Bernoulli}(\theta_{z_i z_j}) \end{aligned}$$

The hyperparameters α_0, β_0, α_1 and β_1 are drawn from a four-dimensional grid where $\alpha_0 + \beta_0$ and $\alpha_1 + \beta_1$ belong to $\{\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8, 16, 32\}$ and $\frac{\alpha_0}{\alpha_0+\beta_0}$ and $\frac{\alpha_1}{\alpha_1+\beta_1}$ belong to

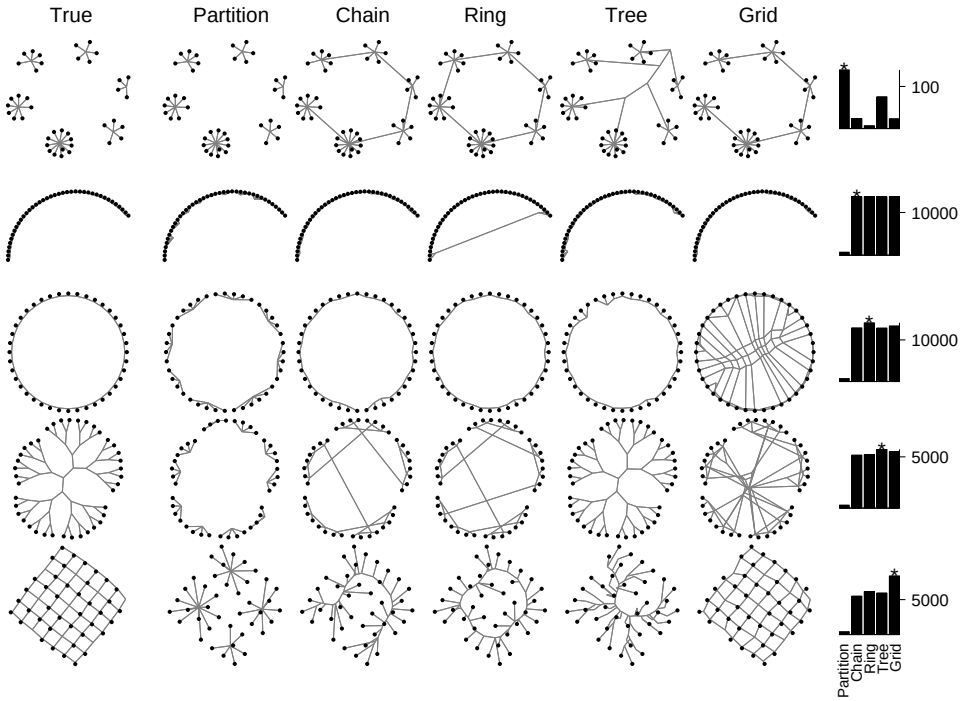

Fig. S5. Structure discovery results for synthetic data. Five sets of features were generated over the graphs in the left column, and five forms were fit to each dataset. The structures found are drawn so that entity positions correspond to positions in the picture of the true structure. Each entity has been connected to the cluster node to which it belongs: for instance, all graphs in the top row have six clusters. The final column shows log posteriors $\log(P(S, F|D))$ for the best structures found, and the best scoring structure is marked with an asterisk. The difference between the scores for the top two structures ranges from 0.63 (indicating that the chain is about twice as likely as the grid on the chain-structured data) to 2245 (indicating that the grid is many orders of magnitude more likely than the ring on the grid-structured data). A constant has been added to the log probabilities along each y axis so that the worst performing structure receives a score close to zero.

$\{0.05, 0.15, \ldots, 0.95\}$. We sample uniformly from all points on this grid where $\frac{\alpha_0}{\alpha_0+\beta_0} \leq \frac{\alpha_1}{\alpha_1+\beta_1}$, which captures the assumption that relation D is most likely to be true of pairs (i, j) that correspond to edges in graph S.

As for the frequency model, we integrate out the parameters:

$$P(D|S) = \sum_{(\alpha_0,\beta_0,\alpha_1,\beta_1)} P(D|S, \alpha_0, \beta_0, \alpha_1, \beta_1)P(\alpha_0, \beta_0, \alpha_1, \beta_1)$$
$$= \sum_{(\alpha_0,\beta_0,\alpha_1,\beta_1)} P(D_0|\alpha_0, \beta_0)P(D_1|\alpha_1, \beta_1)P(\alpha_0, \beta_0, \alpha_1, \beta_1)$$

where D_1 represents the entries in D that correspond to edges in the graph S, and D_0 represents the remaining entries in D. As before, the terms $P(D_0|\alpha_0, \beta_0)$ and $P(D_1|\alpha_1, \beta_1)$ are computed by integrating out θ:

$$P(D_1|\alpha_1, \beta_1) = \int P(D_1|\theta_1)p(\theta_1|\alpha_1, \beta_1)d\theta_1$$

where θ_1 is a vector containing parameters θ_{ab} for all pairs (a, b) such that there is an edge between cluster a and cluster b. $P(D_0|\alpha_0, \beta_0)$ is computed similarly.

Model Implementation

The hierarchical generative model in Fig. 1 can be used for many purposes. If the form of a data set is already known, we can search for the structure S that maximizes $P(S|F)$. If the form of the data is not known, at least two strategies might be tried. For some applications it may be desirable to integrate over the space of structures S and compare forms according to their posterior probabilities $P(F|D)$. Here, however, we search for the structure S and form F that jointly maximize $P(S, F|D)$ (Equation 1). Two considerations motivate this approach. First, we are interested in discovering the structure S that best accounts for the data. Maintaining a posterior distribution over structures may lead to optimal predictions about unobserved features, but human learners often appear to choose just one representation for a problem. Second, even if we wanted to integrate over the space of structures, computing the integral $P(F|D) = \int P(F, S|D)P(S|D)dS$ is a difficult challenge.

Our method for identifying the S and F that maximize $P(S, F|D)$ involves a separate search for each form. Given data D, for each form F we search for the best structure S that is consistent with that form. Since the prior on the space of forms is uniform, the winning structure is the best candidate encountered in any of these searches.

The algorithm used for each of these searches is related to top-down methods for constructing trees and sets of clusters [9, 10], and to the general idea of coarse-to-fine processing [11]. We begin with all the entities in a single cluster, then use graph grammars like those in Fig. 2 to split the entities into multiple clusters. Whenever a cluster node is split, the entities previously assigned to this cluster must be distributed between the two new cluster nodes. We choose two of these entities at random, assign one to each of the new clusters, then go through the remaining entities in a random order, making a greedy assignment for each one. Since this procedure for splitting a cluster node is not deterministic, the search algorithm as a whole is not deterministic. At each iteration, we attempt to split each cluster node several times,

and of all splits considered we accept the candidate that improves the score most. The search is not strictly greedy, since we also use heuristics that attempt to improve the score. One of these heuristics moves entities between cluster nodes, and a second attempts to exchange cluster nodes.

Experiments with synthetic data (Fig. S5) suggest that our search algorithm often recovers the true structure, or a structure very close to the true structure, but we cannot be sure that we have found the best structures for the data sets shown in Figs. 3 and 4. It is possible that improved search algorithms will identify better representations of these data sets.

Features and Similarity

When working with feature data or similarity data, we usually initialize the search process by tying all branch lengths together. Once the score no longer improves, we untie the branch lengths and attempt to improve the score further.

For feature and similarity data, the structures encountered early on in the greedy search can be seen as low-resolution versions of the structure that will eventually be identified as the best. This perspective suggests why a greedy search should often perform well. If we take some true structure and construct a series of representations at increasingly low resolutions, the series should provide a path by which a greedy search can progress from the lowest-resolution version (a structure with all the entities in one cluster) to the true structure.

Relations

A greedy search which moves from low-resolution structures to high-resolution structures should work well when fitting some structural forms (including partitions and dominance hierarchies) to relational data. For other forms, however, a greedy search may fail badly. Consider the case where the true structure is a ring, and each entity sends a link to exactly one other entity. There is no low-resolution version of this structure that seems acceptable: we can group the entities into clusters and organize those clusters into a ring, but the entities in each cluster will tend not to send links to the entities in the next cluster along.

When analyzing relational data, we used two initialization strategies. The first is the same strategy used for feature data: we begin with a graph where all the entities are assigned to a single cluster. The second strategy uses the best clusters found for one of the simplest structural forms: partitions with no self-links (when fitting this form, we initialize the search using the first strategy). These clusters are then used to build initial configurations for each of the remaining structural forms. For example, when searching for rings, we start with a chain that connects the two clusters with the strongest link between them. We continue adding clusters to the ends of this chain until we have a chain including all the clusters, then join the ends of this chain to create the ring that will initialize the greedy search for the best ring structure.

Feature Data

Scores for each form on each data set are shown in Figs. S5, S6 and S7. Since our search algorithm is not deterministic, these figures were generated by repeating each search 10 times and reporting the best structure found.

Given a matrix D with m features, we apply a linear transformation so that the mean value in D is zero, and the maximum entry in $\frac{1}{m}DD^{\mathsf{T}}$ is one. The first property is useful since our model assumes that the features have zero mean. The second property means that it should be sensible to use the same value of the hyperparameter β for both feature and similarity data (we set $\beta = 0.4$).

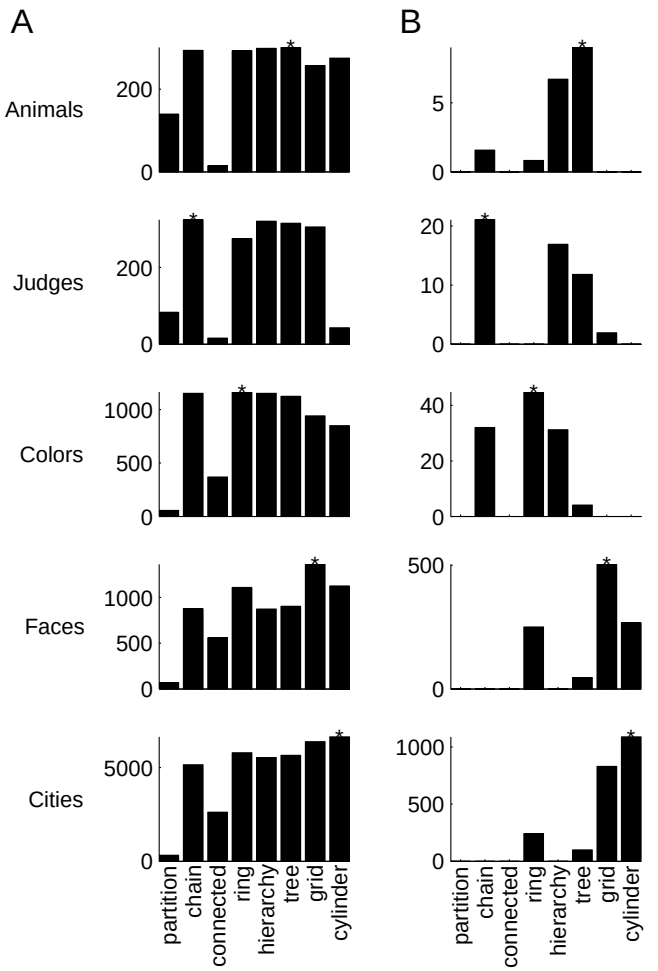

Fig. S6. Scores for eight structural forms on feature and similarity data. (A) Each score represents $\log(P(S, F|D))$ where S is the best structure found for form F. The scores have been translated that the lowest score in each case is close to zero. (B) Relative scores for the top four forms for each data set. The differences between these scores are the same as the differences in A.

If there are missing entries in D, our procedure for transforming the data must be adjusted. In this case, we group the features so that any two features in a given group are observed for precisely the same set of entities. Suppose that the largest group has j features. Consider the reduced matrix $\hat{D}$ that is created by including only these j features, and the entities for which these features are observed. We scale the data so that the mean value in D is zero, and the maximum entry in $\frac{1}{j}\hat{D}\hat{D}^{\mathsf{T}}$ is 1.

[9] In general, we cannot simply ignore the missing data when learning structural forms. If two judges never sat on the same court, there are no features observed for both of them, which encourages the model to assign them to the same node in the structure if their ideological positions are even roughly similar. (Given fully observed data, two entities will usually be assigned to the same node only if they are highly similar.) Groupings of this sort can affect the relative scores of different structural forms. We excluded the first Rehnquist court since Kennedy and Powell (who sat only on that court, and whom Kennedy replaced in 1988) tended to be assigned to the same node, and this grouping appears to be heavily influenced by the fact that these judges never served together.

Synthetic Data

Each synthetic data set contains 40 entities and 2000 features. The features in each data set were generated from the distribution in Equation S3, where $\tilde{\Delta}$ is defined over one of the graphs in the leftmost column of Fig. S5.

Animals

We asked a single participant to make binary judgments indicating whether 106 features applied to 60 animal species. The data include perceptual features (is black), anatomical features (has feet), ecological features (lives in the ocean) and behavioral features (makes loud noises). For the analysis described in the paper we chose 33 species (the species in Fig. 5) that are representative of the full set.

Judges

The Supreme Court data are based on all cases heard between October 1987 and June 2005. This period covers all of the Rehnquist natural courts except the first. Since at most 9 judges voted on any of the cases, the data include many missing entries. We assume that the unobserved entries are missing completely at random, and integrate over all possible values for these entries.[9] The unit of analysis is the case citation (ANALU=0), and we included cases where DEC_TYPE equals 1 or 5 [12]. Voting behaviors were converted to binary values: regular concurrence (3) and special concurrence (4) were converted to majority votes (1), and non-participation (5) was treated as missing data. Any case with a voting behavior other than 1 through 5 was removed from the analysis. The final data set includes 13 judges and 1596 cases.

Similarity Data

When analyzing similarity data, we need to specify an effective number of features m on which the similarity judgments are based. If m is low, then small differences between similarity ratings are likely to be ignored, but if m is high our model will try to account for more of the structure in the data. For all analyses we set $m = 1000$. If a similarity matrix D is not positive semi-definite, we set all negative values in its eigenspectrum to zero, but otherwise apply no pre-processing.

Colors

The Ekman color data were taken from Shepard [13]. Configurations similar to Fig. 3C have been found using multidimensional scaling to locate the colors in two dimensions [13], but a ring provides more appropriate constraints on inductive inference. The ring implies that other pure-wavelength hues will be located somewhere along the ring, but if a two-dimensional configuration were chosen, other hues would be (incorrectly) expected to fall in any region of the space.

Faces

We created 16 stimuli using the FaceGen program [14]. The program includes dimensions for race and gender, and we used four possible values along each dimension. The dissimilarity between faces was defined as the Euclidean distance between their pixel vector representations.

Cities

Dissimilarity was defined as distance along the surface of the earth. Assuming that the earth is spherical, these distances can be calculated using the latitude and longitude of each city.

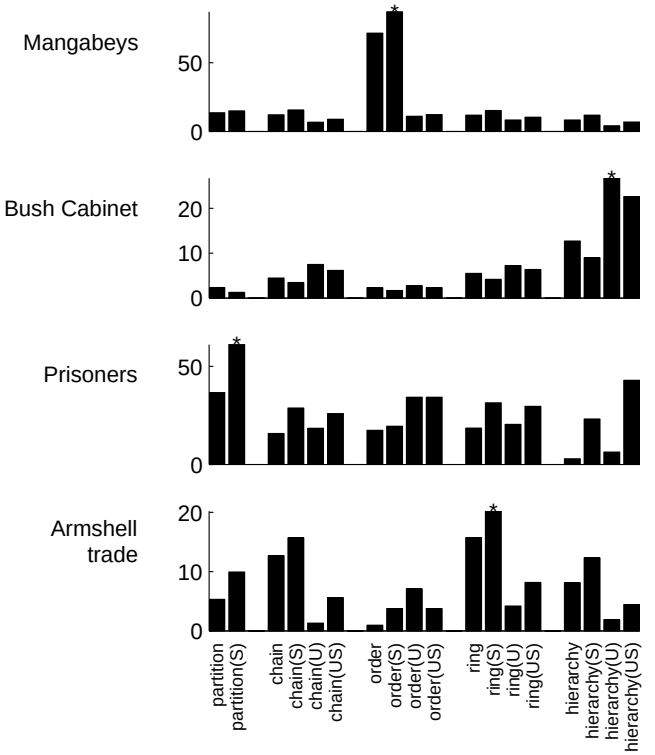

Fig. S7. Scores for eighteen structural forms on relational data. U indicates an undirected form, and S indicates a form with self links (see Fig. S1). The scores have been translated that the lowest score in each case is close to zero.

Relational Data

We used the frequency model to analyze the first two data sets in Fig. 4 and the binary model for the remaining two. We ran our search algorithm 20 times for each (form, data set pair): half of these runs used the first initialization strategy described above, and the remainder used the second strategy.

Mangabeys

The data represent interactions where one animal in a troop of mangabeys submitted to another. Range and Noë [15] consider two types of submissive behavior: in the first, 'the actor jumps or walks away from an approaching individual,' and in the second, 'the actor leans aside or shifts body position in response to another individual that approaches or walks by.' We recoded their data so that a count in the (i, j) cell of the matrix indicates that i caused j to submit.

Bush Cabinet

We ran Google searches on January 26, 2006 to create a matrix D where D_{ij} is the number of hits for the phrase 'i told j,' and i and j vary over 13 members of the Bush administration. Although there are some hits for phrases like 'Bush told Bush,' we set the counts along the diagonal to zero.

Prisoners

The 67 prison inmates were asked 'What fellows in the tier are you closest friends with?' [16] Each inmate mentioned as many friends as he wished. Clique structures similar to Fig. 3C have been discovered by previous clustering algorithms [16], but most of these algorithms assume in advance that the best kind of representation is a set of cliques.

Armshell Trade

Trade relations between 20 New Guinea communities were taken from Hage and Harary [17]. There is a link between i and j if community i sends *mwali* (armshells) to community j.

Modeling Cognitive Development

As children learn more about a domain, their mental representations undergo qualitative transitions that have been likened to paradigm shifts in science [18, 19]. Our model shares this ability to move between qualitatively different representations of a domain. Given a small amount of data, our model may choose a form that is simple, but that does not capture the true structure of the domain. As more data arrive, the model should reach a point where the true structural form is preferred.

To demonstrate a qualitative shift in biological knowledge, we presented our model with increasing numbers of features of the animals in Fig. 3*A*. We could have run this simulation by randomly sampling smaller data sets from the full feature matrix, but the results might have been influenced by idiosyncratic properties of the small data sets sampled. To avoid this problem, we directly specified the covariance of each data set, and worked with the similarity version of our model. We analyzed data sets where the effective number of features was 5, 20, or 110, and the similarity matrix in each case was the covariance matrix for the full set of animal features. Even though the similarity matrices are identical, increasing the effective number of features should allow the model to discover more complex representations. When only 5 features are provided, the model should attempt only to fit the broad trends in the data, but given 110 features, the model should attempt to explain some of the more subtle variation in the data.

Fig. 5 shows the representations chosen by our model for each data set. At first, the simplest form is preferred, and the model chooses a set of clusters. Given 20 features, the tree form is preferred, but the chosen tree is simpler than the tree in Fig. 3*A*. The final tree is identical to the tree in Fig. 3*A*: note that a similarity data set with 110 features is effectively identical to the data set that led to Fig. 3*A*.

The developmental shift in Fig. 5 appears similar to a trajectory that children follow as they learn the meanings of words. Early in development, children appear to respect the assumption of mutual exclusivity: they organize objects into a set of non-overlapping clusters, with one category label allowed per cluster [20]. Eventually, however, children realize that objects can be organized into taxonomic hierarchies. Fig. 5 suggests that this insight may be driven in part by the amount of data available to a word learner.

The ability to learn from raw data may support some of the earliest and most fundamental shifts in children's thinking. Bottom-up learning, however, can only explain some aspects of cognitive development, and explicit instruction may contribute to the majority of developmental shifts once children have become proficient language users. Although we have focused on learning representations from raw data, hierarchical approaches like ours can naturally handle linguistic input at multiple levels of abstraction, including all three levels in Fig. 1*A*. Linguistic input can provide new features (e.g. 'whales breathe air'), and can also provide direct information about a structure S (e.g. 'whales belong with the mammals rather than the fish') or a form F (e.g. 'the theory of evolution implies that animals should be organized into a tree'). Modeling learning when input is simultaneously provided at several levels of abstraction is an important goal for future work.

Related Work

In statistical terms, our method for discovering structural forms can be viewed as an instance of model selection [21]. From a Bayesian perspective, model selection can be achieved by describing a hypothesis space of models (for us, each model is a pair (S, F)) and using Bayesian inference to choose between them. Other approaches are sometimes proposed: Pruzansky et al. [22] decide whether a similarity matrix is better described by a tree or a two dimensional space by finding the best instance of each form and choosing the structure that accounts for the most variance. Several authors [23, 24] have proposed methods for distinguishing between cluster structures and dimensional structures.

A key feature of our Bayesian approach is that it automatically penalizes unnecessarily complex models. Some such penalty is essential when considering structural forms of different complexities, since complex forms (e.g. fully connected graphs) can easily mimic simpler forms. Each chain, for example, is a special case of a grid, and it follows that the best grid S_g will account for any data set D at least as well as the best chain S_c: $P(D|S_g) \geq P(D|S_c)$. The approach of Pruzansky et al. [22] will therefore never choose the simpler form, even when the data D were actually generated over a chain.[10]

Bayesian model selection has previously been used to learn models that are only as complex as warranted by the data, but often the structural form of the model is assumed to be known in advance. For instance, Bayesian methods can identify the number of clusters in a mixture model [25], or the number of dimensions in a spatial model [26]. Bayesian methods have also occasionally been used to control complexity in hybrid models with two different kinds of representations, such as discrete features and spatial dimensions [27]. Compared to previous learning algorithms that rely on statistical model selection, two aspects of our approach are particularly distinctive. First, we formulate the problem of structure discovery as an inference in a hierarchical model where the structural form of the domain and a specific graph structure are both represented as latent variables. Second, we specify and search a diverse set of structural forms using grammars for growing graph-structured probabilistic models.

Feature Data

Our model for feature data grows out of previous work on learning the structure of graphical models [5, 6, 7]. Previous models usually belong to one of two families. The first family includes models that impose no strong constraints on the form of the graph structures that are learned. Bayesian approaches within this family generally use a prior that includes all possible graph structures, and the prior over this space is usually relatively simple—for example, Dobra et al. (2004) use a prior that favors graphs with small numbers of edges. Models in the second family assume strong constraints on the form of the

[10] Pruzansky et al. [22] recognize the importance of model complexity, and justify their approach by arguing that the complexity of trees is approximately equal to the complexity of two dimensional spaces.

graph to be discovered, but these constraints are fixed from the start, not learned from data. Approaches in this second family include algorithms for phylogenetic reconstruction [28] that attempt to discover tree-structured graphical models.

Our approach falls in the little-explored territory between these two families of models. Instead of working with generic priors over the set of all possible graph structures, our approach concentrates the prior probability mass on graphs that correspond to one of a small number of structural forms.[11] The ultimate argument for such a prior is that it provides inductive constraints [29] that are well-matched to the problems we wish to solve. The need for inductive constraints is most pressing when dealing with sparse data, and sparse data are the rule rather than the exception in both cognitive development and scientific discovery.

Inferences about novel entities account for some of the most common cases where the available data are sparse. Consider, for example, two children who both have tree-structured representations of a set of familiar species. Suppose that the first child realizes that living kinds are tree-structured, but that the second child does not—in other words, suppose that the second child entertained all possible graph structures, and just happened to settle on one that was tree structured. Imagine, now, that both children encounter a new animal. The first child can slot the animal into her tree relatively easily—she knows, for example, that the new species will attach to the taxonomy at exactly one point. The second child faces a much more difficult problem. Since she need not preserve the tree structure of her current representation, there may be many edges that join the new species to her current representation, and deciding which of these edges exist may require a large amount of data.

Relational Data

Our relational model also builds on previous methods for discovering structure in relational data [30, 31, 32, 33]. Consider, for instance, the many previous models for relational clustering, or identifying clusters of entities that relate to each other in predictable ways. As for the feature-based case, previous approaches to relational clustering usually belong to one of two families. The first family includes models that impose no strong constraint on the form of the structures to be discovered. Stochastic blockmodels [34, 35] are one example: they do not incorporate the notion of structural form, and cannot explicitly realize when a set of clusters takes a simple form like a ring, or a set of cliques. The second family includes models that assume that the structural form is known in advance. For example, there are several algorithms for discovering community structures in networks [33, 36]. These approaches usually assume that the data are organized into a set of cliques, and that individuals from any given clique tend only to be related to others from the same clique.

Our model again occupies the little-explored territory between these two families of approaches. Structural forms are useful because they provide strong inductive constraints, and the ability to discover these constraints allows a learner to efficiently handle novel inductive contexts. To see the importance of structural form in the relational setting, consider a relational analogue of the novel species scenario described earlier. Suppose that two baboons have similar representations of the interactions between animals in their troop—representations that take the form of an order. One baboon realizes the structural form of the representation, and the other has independently memorized the edges in the representation. Suppose now that a new baboon appears, and dominates the baboon that used to occupy the first place in the order. The baboon who knows the structural form of the group can predict that the new baboon will dominate all the other animals, but the baboon who has memorized edges can come to no strong conclusion—for her, any set of directed edges may join the new baboon to the remaining animals in the troop.

1. Nagl M (1986) Set theoretic approaches to graph grammars. *Proceedings of the 3rd International Workshop on Graph-grammars and their application to computer science* (Springer-Verlag, London, UK), pp 41–54.
2. Escobar MD, West M (1995) Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* 90:577–588.
3. Neal R (2003) Density modeling and clustering using Dirichlet diffusion trees. *Bayesian Statistics 7*, eds Bernardo, JM et al. (Oxford University Press, Oxford), pp 619–629.
4. Zhu X, Lafferty J, Ghahramani Z (2003) Semi-supervised learning: from Gaussian fields to Gaussian processes, (Carnegie-Mellon University), Technical Report CMU-CS-03-175.
5. Dempster AP (1972) Covariance selection. *Biometrics* 28:157–175.
6. Whittaker J (1990) *Graphical models in applied multivariate statistics* (Wiley, Chichester, UK).
7. Dobra A, Jones B, Hans C, Nevins J, West M (2004) Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis* 90:196–212.
8. Ackley D, Hinton G, Sejnowski T (1985) A learning algorithm for Boltzmann machines. *Cognitive Science* 9:147–169.
9. Boley DL (1998) Principal direction divisive partitioning. *Data mining and knowledge discovery* 2:325–344.
10. Manning CD, Raghavan P, Schütze H (2008) *Introduction to Information Retrieval* (Cambridge University Press, Cambridge).
11. Gangaputra S, Geman D (2006) A design principle for coarse-to-fine classification. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* pp 1877–1884.
12. Spaeth HJ (2005) *United States Supreme Court judicial database, 1953-2005 Terms.*
13. Shepard RN (1980) Multidimensional scaling, tree-fitting, and clustering. *Science* 210:390–398.
14. FaceGen modeller from Singular Inversions Inc.
15. Range F, Noë R (2002) Familiarity and dominance relations among female sooty mangabeys in the Tai national park. *American Journal of Primatology* 56:137–153.
16. MacRae, J (1960) Direct factor analysis of sociometric data. *Sociometry* 22:360–371.
17. Hage P, Harary F (1991) *Exchange in Oceania: A graph theoretic analysis* (Oxford University Press, Oxford).
18. Carey S (1985) *Conceptual change in childhood* (MIT Press, Cambridge, MA).
19. Kuhn TS (1970) *The structure of scientific revolutions* (University of Chicago Press, Chicago), 2nd edition.
20. Markman E (1989) *Naming and categorization in children* (MIT Press, Cambridge, MA).
21. Kass RE, Raftery AE (1995) Bayes factors. *Journal of the American Statistical Association* 90:773–795.
22. Pruzansky S, Tversky A, Carroll JD (1982) Spatial versus tree representations of proximity data. *Psychometrika* 47:3–19.
23. Waller NG, Meehl PE (1998) *Multivariate taxometric procedures: distinguishing types from continua* (Sage, Thousand Oaks, CA).
24. Boeck PD, Wilson M, Acton GS (2005) A conceptual and psychometric framework for distinguishing categories and dimensions. *Psychological Review* 112:129–158.

[11] Even though the notion of structural form is the most distinctive feature of our approach, our work differs from previous structure learning models in at least three other respects. First, standard methods for learning the structure of Gaussian graphical models do not allow latent nodes. Second, these methods make no attempt to cluster the nodes. Third, these methods allow graphs where some of the edges capture negative covariances. For the generative model in Equation S3, an edge between two entities always encourages the entities to have similar feature values.

25. Green P (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82:711–732.

26. Bishop CM (2002) Bayesian PCA. *Advances in Neural Information Processing Systems 11* pp 382–388.

27. Navarro DJ, Lee MD (2002) Combining dimensions and features in similarity-based representations. *Advances in Neural Information Processing Systems 15* pp 59–66.

28. Huelsenbeck JP, Ronquist F (2001) MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics* 17:754–755.

29. Mitchell TM (1997) *Machine learning* (McGraw Hill, New York).

30. White HC, Boorman SA, Breiger RL (1976) Social structure from multiple networks. *American Journal of Sociology* 81:730–780.

31. Nowicki K, Snijders TAB (2001) Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association* 96:1077–1087.

32. Taskar B, Segal E, Koller D (2001) Probabilistic classification and clustering in relational data. *Proceedings of the 17th International Joint Conference on Artificial Intelligence* pp 870–876.

33. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99:7821–7826.

34. Wang YJ, Wong GY (1987) Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association* 82:8–19.

35. Kemp C, Tenenbaum JB, Griffiths TL, Yamada T, Ueda N (2006) Learning systems of concepts with an infinite relational model. *Proceedings of the 21st National Conference on Artificial Intelligence*.

36. Kubica J, Moore A, Schneider J, Yang Y (2002) Stochastic link and group detection. *Proceedings of the 17th National Conference on Artificial Intelligence* pp 798–804.