How to Grow a Mind:
Statistics, Structure and Abstraction

Josh Tenenbaum
MIT Department of Brain and Cognitive Sciences
CSAIL

Funding: AFOSR, ONR, ARL, DARPA, James S. McDonnell Foundation, NTT, Google, Schlumberger, Shell, Paul Newton Chair
Tom Griffiths  Charles Kemp  Amy Perfors  Noah Goodman

Vikash Mansinghka  Chris Baker  Russ Salakhutdinov  Peter Battaglia

Fei Xu  Owain Evans  Andreas Stuhlmuller
Owen Macindoe  Dan Roy  David Wingate
Brenden Lake  Jess Hamrick  Steve Piantadosi
Tomer Ullman  Lauren Schmidt  Steve Piantadosi
The goal

“Reverse-engineering the mind”

Understand human learning and inference in our best engineering terms, and use that knowledge to build more human-like machine learning and inference systems.
The big question

How does the mind get so much out of so little?

Our minds build rich models of the world and make strong generalizations from input data that is sparse, noisy, and ambiguous – in many ways far too limited to support the inferences we make.

How do we do it?
Learning words for objects
Learning words for objects

“tufa”

“tufa”

“tufa”
The big question

How does the mind get so much out of so little?

– Perceiving the world from sense data
– Learning about kinds of objects and their properties
– Learning the meanings of words, phrases, and sentences
– Inferring causal relations
– Learning and using intuitive theories of physics, psychology, biology, social structure…
Southgate and Csibra, 2009

Heider and Simmel, 1944
The approach: *learning with knowledge*

1. How does abstract knowledge guide learning and inference from sparse data?
   
   Bayesian inference in probabilistic generative models.
   
   $$ P(h|d) = \frac{P(d|h)P(h)}{\sum_{h_i \in H} P(d|h_i)P(h_i)} $$

2. What form does abstract knowledge take, across different domains and tasks?
   
   Probabilities defined over a range of structured representations: spaces, graphs, grammars, predicate logic, schemas, programs.

3. How is abstract knowledge itself acquired – balancing complexity versus fit, constraint versus flexibility?
   
   Hierarchical models, with inference at multiple levels ("learning to learn"). Nonparametric ("infinite") models, growing complexity and adapting their structure as the data require.
Perception as Bayesian inference

Weiss, Simoncelli & Adelson (2002): “Slow and smooth” priors

Kording & Wolpert (2004): Priors in sensorimotor integration
Perception as Bayesian inference

Wainwright, Schwartz & Simoncelli (2002): Bayesian ideal observers based on natural scene statistics

Does this approach extend to cognition?
Everyday prediction problems
(Griffiths & Tenenbaum, *Psych. Science* 2006)

- You read about a movie that has made $60 million to date. How much money will it make in total?
- You see that something has been baking in the oven for 34 minutes. How long until it’s ready?
- You meet someone who is 78 years old. How long will they live?
- Your friend quotes to you from line 17 of his favorite poem. How long is the poem?
- You meet a US congressman who has served for 11 years. How long will he serve in total?
- You encounter a phenomenon or event with an unknown extent or duration, $t_{total}$, at a random time or value of $t < t_{total}$. What is the total extent or duration $t_{total}$?
Priors $P(t_{total})$ based on empirically measured durations or magnitudes for many real-world events in each class:

Median human judgments of the total duration or magnitude $t_{total}$ of events in each class, given one random observation at a duration or magnitude $t$, versus Bayesian predictions (median of $P(t_{total}|t)$).
Learning words for objects

"tufa"

What is the right prior?
What is the right hypothesis space?
How do learners acquire that background knowledge?
Learning words for objects

“tufa”

(Collins & Quillian, 1969)

(Kiani et al., 2007, IT population responses; c.f. Hung et al., 2005)
Learning words for objects

Bayesian inference over tree-structured hypothesis space:

(Xu & Tenenbaum, *Psych. Review* 2007; Schmidt & Tenenbaum, in prep)
Learning to learn words
(w/ Kemp, Perfors)

• Learning which features count for which kinds of concepts and words.

  Show me the dax…

  This is a dax.

  – *Shape bias* (Smith) for simple solid objects (2 years).
  – *Material bias* for non-solid substances (~3 years).
  – ...

• Learning the form of structure in a domain.
  – Early hypotheses follow *mutual exclusivity* (Markman).
    A tree-structured hierarchy of nameable categories emerges only later.
Learning to learn: which object features count for word learning?

Query image

Retrieved images with learned metric

Retrieved images with nearest neighbours

46,875 “texture of textures” features:

[Salakhutdinov, Tenenbaum, Torralba ‘10]
Learning to learn: which object features count for word learning?

Query image

Retrieved images with learned metric

Retrieved images with nearest neighbours

46,875 “texture of textures” features:

“Similar categories have similar similarity metrics”

[Salakhutdinov, Tenenbaum, Torralba ‘10]
Learning to learn: which object features count for word learning?

46,875 “texture of textures” features:

“Similar categories have similar similarity metrics”

[Salakhutdinov, Tenenbaum, Torralba ‘10]
Learning to learn: which object features count for word learning?

Tree learned with nCRP prior

[Salakhutdinov, Tenenbaum, Torralba ‘10]
Learning to learn: which object features count for word learning?

Tree learned with nCRP prior

MSR dataset:

- aeroplanes
- benches and chairs
- bicycles/single
- cars/front
- cars/rear
- cars/side
- signs
- buildings
- chimneys
- doors'
- scenes/office
- scenes/urban
- windows
- trees
- birds
- flowers
- leaves
- scenes/countryside
- animals/cows
- animals/sheep
- clouds

Learned metric: Euclidean distance
Oracle (best possible metric)

ROC Curve for 1-shot learning

[Salakhutdinov, Tenenbaum, Torralba ‘10]
HDP-RBM
[Salakhutdinov, Tenenbaum, Torralba, in prep]

High-level class-sensitive features
[HDP topic model (admixture)]

*learned from 100 CIFAR classes*

Low-level general features
[Restricted Boltzmann Machine]

*learned from 4 million tiny images*

Images
(= 32 x 32 pixels x 3 RGB)

\[
\begin{align*}
X_{ij} & \quad \text{1000 units} \\
Y_{ij} & \quad \text{3072 units}
\end{align*}
\]
HDP-RBM
[Salakhutdinov, Tenenbaum, Torralba, in prep]

Learned tree structure of classes [nested CRP prior]

High-level class-sensitive features
[HDP topic model (admixture)]

learned from 100 CIFAR classes

Low-level general features
[Restricted Boltzmann Machine]

learned from 4 million tiny images

Images
(= 32 x 32 pixels x 3 RGB)
The characters challenge
("MNIST++" or "MNIST*")
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge
("MNIST++" or "MNIST*")
The characters challenge
(“MNIST++” or “MNIST*”)

[Image of handwritten characters]
The characters challenge
("MNIST++" or "MNIST*")
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge
(“MNIST++” or “MNIST*”)

The characters challenge
(“MNIST++” or “MNIST*”)

[Image of handwritten digits]
The characters challenge
("MNIST++" or "MNIST*")
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge
(“MNIST++” or “MNIST*”)
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge
(“MNIST++” or “MNIST*”)
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge
("MNIST++" or "MNIST*")
The characters challenge
("MNIST++" or "MNIST*")
The characters challenge
("MNIST++" or "MNIST*")
The characters challenge
("MNIST++" or "MNIST*")
The characters challenge
(“MNIST++” or “MNIST*”)

[Image of handwritten characters]
The characters challenge
(“MNIST++” or “MNIST*”)

[Image of ancient script]
The characters challenge
(“MNIST++” or “MNIST*”)
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge
(“MNIST++” or “MNIST*”)
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge
(“MNIST++” or “MNIST*”)
The characters challenge
(“MNIST++” or “MNIST*”)
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge ("MNIST++" or "MNIST*")
The characters challenge
(“MNIST++” or “MNIST*”)

The image contains a grid of handwritten characters, typical of the MNIST dataset, which is commonly used for training handwriting recognition algorithms.
Learned features

Low-level general-purpose features from RBM

High-level class-sensitive features from HDP (composed of RBM features)
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model
fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model fantasies
Model

fantasies
Model fantasies
Learning from very few examples

3 examples of a new class

Conditional samples in the same class

Inferred super-class
Learning from very few examples
Learning from very few examples
Learning from very few examples
Learning from very few examples
Learning from very few examples
Learning from very few examples
Learning from very few examples
Learning from very few examples
Area under ROC curve for same/different
(1 new class vs. 1000 distractor classes)

 Pixels  LDA-RBM (unsupervised)  LDA-RBM (class conditional)  HDP-RBM (flat)  HDP-RBM (tree)

[Averaged over 50 test classes]
Learning to learn: what is the right form of structure for the domain?
Learning to learn: what is the right form of structure for the domain?

People can discover structural forms…

– Children
  
e.g., hierarchical structure of category labels, cyclical structure of seasons or days of the week, clique structure of social networks.

– Scientists

  Linnaeus
  Kingdom Animalia
  Phylum Chordata
  Class Mammalia
  Order Primates
  Family Hominidae
  Genus Homo
  Species *Homo sapiens*

  Darwin

  … but standard learning algorithms assume fixed forms.

  – Hierarchical clustering: tree structure
  – $k$-means clustering, mixture models: flat partition
  – Principal components analysis: low-dimensional spatial structure
Goal: A universal framework for unsupervised learning

“Universal Learner”

Data → K-Means
Hierarchical clustering
Factor Analysis
PCA
Manifold learning
Circumplex models
...

→ Representation
Hypothesis space of structural forms
(Kemp & Tenenbaum, PNAS 2008)
A hierarchical Bayesian approach
(Kemp & Tenenbaum, PNAS 2008)

\[ P(F) \]

\( F: \text{form} \)

\[ P(S \mid F) \]

\( S: \text{structure} \)

\[ P(D \mid S) \]

\( D: \text{data} \)

\[ P(S, F \mid D) \propto P(D \mid S)P(S \mid F)P(F) \]
A hierarchical Bayesian approach
(Kemp & Tenenbaum, PNAS 2008)

\[ P(F) \]

\( F: \) form

\[ P(S \mid F) \]

Simplicity
(Bayes Occam’s razor)

\[ P(D \mid S) \]

Fit to data
(Smoothness: Gaussian process based on graph Laplacian)

\( S: \) structure

\[ D: \) data \]

\[ \Sigma = \tilde{\Delta}^{-1}(S) \]

\[ P(S, F \mid D) \propto P(D \mid S) P(S \mid F) P(F) \]
Development of structural forms as more data are observed

5 features

110 features

20 features

“blessing of abstraction”
Understanding intelligence requires us to go beyond the statistician’s toolkit: Inference over fixed sets of random variables, linked by simple (or well-understood) distributions.

“Probabilistic programming” (NIPS ’08 workshop): Machine learning and Probabilistic AI must expand to include the full computer science toolkit.

- Inference over flexible data structures.
- Complex generative models based on stochastic programs, to capture the rich causal texture of the world.
The Infinite PCFG using Hierarchical Dirichlet Processes

Percy Liang        Slav Petrov        Michael I. Jordan        Dan Klein
Computer Science Division, EECS Department
University of California at Berkeley
Berkeley, CA 94720
{pliang, petrov, jordan, klein}@cs.berkeley.edu

HDP-PCFG

\[ \beta \sim \text{GEM}(\alpha) \]  \quad \text{[draw top-level symbol weights]}  

For each grammar symbol \( z \in \{1, 2, \ldots \} \):

\[ \phi_z^T \sim \text{Dirichlet}(\alpha_z^T) \]  \quad \text{[draw rule type parameters]}  
\[ \phi_z^E \sim \text{Dirichlet}(\alpha_z^E) \]  \quad \text{[draw emission parameters]}  
\[ \phi_z^B \sim \text{DP}(\alpha_z^B, \beta \beta^T) \]  \quad \text{[draw binary production parameters]}  

For each node \( i \) in the parse tree:

\[ t_i \sim \text{Multinomial}(\phi_{z_i}^T) \]  \quad \text{[choose rule type]}  
If \( t_i = \text{Emission} \):

\[ x_i \sim \text{Multinomial}(\phi_{z_i}^E) \]  \quad \text{[emit terminal symbol]}  
If \( t_i = \text{Binary-Production} \):

\[ (z_{L(i)}, z_{R(i)}) \sim \text{Multinomial}(\phi_{z_i}^B) \]  \quad \text{[generate children symbols]}  

Figure 2: The definition and graphical model of the HDP-PCFG. Since parse trees have unknown structure, there is no convenient way of representing them in the visual language of traditional graphical models. Instead, we show a simple fixed example tree. Node 1 has two children, 2 and 3, each of which has one observed terminal child. We use \( L(i) \) and \( R(i) \) to denote the left and right children of node \( i \).
Intuitive psychology

Southgate and Csibra, 2009

Heider and Simmel, 1944
Modeling human action understanding

- Latent mental states: beliefs and desires.
- Principle of rationality: Assume that other agents will tend to take sequences of actions that most effectively achieve their desires given their beliefs.
- Model this more formally as Bayesian inference?

\[
p(B, D \mid A) \propto p(A \mid B, D) \, p(B, D)
\]
Modeling human action understanding

• Latent mental states: *beliefs* and *desires*.

• Principle of rationality: Assume that other agents will tend to take sequences of actions that most effectively achieve their desires given their beliefs.

• *Bayesian inverse planning* in a Partially Observable Markov Decision Process (MDP).
  
  (c.f. inverse optimal control, inverse RL)

\[
p(B, D \mid A) \propto p(A \mid B, D) \ p(B, D)
\]
Goal inference as inverse probabilistic planning
(Baker, Tenenbaum & Saxe, Cognition, 2009)
Theory of mind: Joint inferences about beliefs and preferences
(Baker, Saxe & Tenenbaum, in prep)

Food truck scenarios:
Intuitive physics
Modeling intuitive physical inferences about visual scenes

(Battaglia, Hamrick, Tenenbaum, Torralba, Wingate)

1. “Vision as inverse graphics.”
   - Recover a physically realistic 3D scene description by Bayesian inference in a probabilistic rendering model.

2. “Physics as forward physics.”
   - Run forward simulations with probabilistic Newtonian mechanics. (Cf. Griffiths, Sanborn, Mansinghka)
     • Starting point: dynamics are fundamentally deterministic; uncertainty enters from imperfect state estimates by vision.
     • Next steps: uncertainty about mechanics, simulation noise, noise in working memory.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>tower id</td>
<td>26</td>
<td>0.022</td>
</tr>
<tr>
<td>response</td>
<td>0.569</td>
<td>+/- 0.022</td>
</tr>
<tr>
<td>response time</td>
<td>2.605</td>
<td>+/- 0.015</td>
</tr>
<tr>
<td>stability</td>
<td>0.49</td>
<td>+/- 0.05</td>
</tr>
<tr>
<td>displacement</td>
<td>-0.461</td>
<td>+/- 0.028</td>
</tr>
<tr>
<td>num falling blocks</td>
<td>-0.541</td>
<td>+/- 0.048</td>
</tr>
<tr>
<td>num samples</td>
<td>0.9</td>
<td>+/- 0.247</td>
</tr>
<tr>
<td>time to fall</td>
<td>0.239</td>
<td>+/- 0.1</td>
</tr>
<tr>
<td>Movement threshold</td>
<td>0.3</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Evidence threshold</td>
<td>5.0</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300.0</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Belief noise</td>
<td>0.4</td>
<td>+/- 0.0</td>
</tr>
</tbody>
</table>

Physics is: OFF
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>tower id</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>response</td>
<td>0.173</td>
<td>+/- 0.02</td>
</tr>
<tr>
<td>response time</td>
<td>2.718</td>
<td>+/- 0.023</td>
</tr>
<tr>
<td>stability</td>
<td>0.201</td>
<td>+/- 0.047</td>
</tr>
<tr>
<td>displacement</td>
<td>-0.103</td>
<td>+/- 0.04</td>
</tr>
<tr>
<td>num falling blocks</td>
<td>-0.182</td>
<td>+/- 0.046</td>
</tr>
<tr>
<td>num samples</td>
<td>1.982</td>
<td>+/- 0.282</td>
</tr>
<tr>
<td>time to fall</td>
<td>0.101</td>
<td>+/- 0.079</td>
</tr>
<tr>
<td>Movement threshold</td>
<td>0.3</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Evidence threshold</td>
<td>5.0</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300.0</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Belief noise</td>
<td>0.4</td>
<td>+/- 0.0</td>
</tr>
</tbody>
</table>

Physics is: OFF
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>tower id</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>response</td>
<td>-1.319</td>
<td>+/- 0.028</td>
</tr>
<tr>
<td>response time</td>
<td>1.872</td>
<td>+/- 0.121</td>
</tr>
<tr>
<td>stability</td>
<td>-1.625</td>
<td>+/- 0.05</td>
</tr>
<tr>
<td>displacement</td>
<td>3.148</td>
<td>+/- 0.185</td>
</tr>
<tr>
<td>num falling blocks</td>
<td>1.654</td>
<td>+/- 0.05</td>
</tr>
<tr>
<td>num samples</td>
<td>-1.68</td>
<td>+/- 0.1</td>
</tr>
<tr>
<td>time to fall</td>
<td>-0.575</td>
<td>+/- 0.062</td>
</tr>
<tr>
<td>Movement threshold</td>
<td>0.3</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Evidence threshold</td>
<td>5.0</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300.0</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Belief noise</td>
<td>0.4</td>
<td>+/- 0.0</td>
</tr>
</tbody>
</table>

Physics is: OFF
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>±</th>
</tr>
</thead>
<tbody>
<tr>
<td>tower id</td>
<td>18</td>
<td>±0.029</td>
</tr>
<tr>
<td>response</td>
<td>-1.001</td>
<td>±0.029</td>
</tr>
<tr>
<td>response time</td>
<td>2.45</td>
<td>±0.084</td>
</tr>
<tr>
<td>stability</td>
<td>-1.268</td>
<td>±0.061</td>
</tr>
<tr>
<td>displacement</td>
<td>0.899</td>
<td>±0.061</td>
</tr>
<tr>
<td>num falling blocks</td>
<td>1.289</td>
<td>±0.059</td>
</tr>
<tr>
<td>num samples</td>
<td>-0.767</td>
<td>±0.124</td>
</tr>
<tr>
<td>time to fall</td>
<td>-0.447</td>
<td>±0.072</td>
</tr>
<tr>
<td>Movement threshold</td>
<td>0.3</td>
<td>±0.0</td>
</tr>
<tr>
<td>Evidence threshold</td>
<td>5.0</td>
<td>±0.0</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300.0</td>
<td>±0.0</td>
</tr>
<tr>
<td>Belief noise</td>
<td>0.4</td>
<td>±0.0</td>
</tr>
</tbody>
</table>

Physics is: OFF
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>tower id</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>response</td>
<td>-0.112</td>
<td>+/- 0.007</td>
</tr>
<tr>
<td>response time</td>
<td>2.787</td>
<td>+/- 0.015</td>
</tr>
<tr>
<td>stability</td>
<td>-0.338</td>
<td>+/- 0.048</td>
</tr>
<tr>
<td>displacement</td>
<td>0.062</td>
<td>+/- 0.044</td>
</tr>
<tr>
<td>num falling blocks</td>
<td>0.281</td>
<td>+/- 0.047</td>
</tr>
<tr>
<td>num samples</td>
<td>0.599</td>
<td>+/- 0.178</td>
</tr>
<tr>
<td>time to fall</td>
<td>-0.964</td>
<td>+/- 0.042</td>
</tr>
<tr>
<td>Movement threshold</td>
<td>0.3</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Evidence threshold</td>
<td>5.0</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300.0</td>
<td>+/- 0.0</td>
</tr>
<tr>
<td>Belief noise</td>
<td>0.4</td>
<td>+/- 0.0</td>
</tr>
</tbody>
</table>

Physics is: OFF
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>tower id</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>response</td>
<td>-0.673</td>
<td>+/- 0.038</td>
</tr>
<tr>
<td>response time</td>
<td>2.522</td>
<td>+/- 0.053</td>
</tr>
<tr>
<td>stability</td>
<td>-0.5</td>
<td>+/- 0.058</td>
</tr>
<tr>
<td>displacement</td>
<td>0.272</td>
<td>+/- 0.054</td>
</tr>
<tr>
<td>num falling blocks</td>
<td>0.508</td>
<td>+/- 0.052</td>
</tr>
<tr>
<td>num samples</td>
<td>-0.199</td>
<td>+/- 0.177</td>
</tr>
<tr>
<td>time to fall</td>
<td>0.16</td>
<td>+/- 0.08</td>
</tr>
<tr>
<td>Movement threshold</td>
<td>0.3</td>
<td>+/- 0</td>
</tr>
<tr>
<td>Evidence threshold</td>
<td>5.0</td>
<td>+/- 0</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300.0</td>
<td>+/- 0</td>
</tr>
<tr>
<td>Belief noise</td>
<td>0.4</td>
<td>+/- 0</td>
</tr>
</tbody>
</table>

Physics is: OFF
Stability inferences

Mean human stability judgment

Model prediction (expected proportion of tower that will fall)
Intuitive physics in infants

(Teglas, Vul, Gonzalez, Girotto, Tenenbaum, Bonatti, under review)
Probabilistic programming languages

Universal language for describing generative models +
generic tools for (approximate) probabilistic inference.

• Probabilistic logic programming (Prolog)
  – BLOG (Russell, Milch et al)
  – Markov Logic (Domingos et al)
  – ICL (Poole)

• Probabilistic functional programming (lisp) or
  imperative programming (Matlab)
  – Church: stochastic lisp (Goodman, Mansinghka et al)
  – Monte™ (Mansinghka & co. @ Navia Systems)
  – Stochastic Matlab (Wingate)
  – IBAL: probabilistic ML (Pfeffer)
  – HANSEI: probabilistic OCaml (Oleg, Shan)
Learning as program induction, cognitive development as program synthesis

• Ultimately would like to understand development of intuitive psychology, intuitive physics as program synthesis.

• Shorter-term goals & warm-up problems:
  – Graph grammars for structural form. [Kemp & Tenenbaum]
  – Motor programs for handwritten characters. [Revow, Williams, Hinton; Lake, Salakhutdinov, Tenenbaum]
  – Learning functional aspects of language: determiners, quantifiers, prepositions, adverbs. [Piantadosi, Goodman Tenenbaum; Liang et al.; Zettlemoyer et al., …]
Conclusions

How does the mind get so much from so little, in learning about objects, categories, causes, scenes, sentences, thoughts, social systems?

A toolkit for studying the nature, use and acquisition of abstract knowledge:

- *Bayesian inference* in probabilistic generative models.
- Probabilistic models defined over a range of *structured representations*: spaces, graphs, grammars, predicate logic, schemas, and other data structures.
- *Hierarchical models*, with inference at multiple levels of abstraction.
- *Nonparametric models*, adapting their complexity to the data.
- Learning and reasoning in *probabilistic programming languages*.

An alternative to classic “either-or” dichotomies: “Nature” versus “Nurture”, “Logic” (Structure, Rules, Symbols) versus “Probability” (Statistics).

- How can domain-general mechanisms of learning and representation build domain-specific abstract knowledge?
- How can structured symbolic knowledge be acquired by statistical learning?

A different way to think about the development of a cognitive system.

- Powerful abstractions can be learned surprisingly quickly, together with or prior to learning the more concrete knowledge they constrain.
- Structured symbolic representations need not be rigid, static, hand-wired, brittle. Embedded in a probabilistic framework, they can grow dynamically and robustly in response to the sparse, noisy data of experience.
How could this work in the brain?
The “sampling hypothesis”

Hinton, Dayan, Pouget, Zemel, Schrater, Lengyel, Fiser, Berkes, Griffiths, Steyvers, Vul, Goodman, Tenenbaum, Gershman, ...

The “sampling hypothesis”

Marr’s levels

Computational

Algorithmic

Neural

Particle filtering

Importance sampling

Markov Chain Monte Carlo (MCMC)

(d)efine (occurs a t)
(or (spontaneous a t))
(do a t)
(fold (lambda (x y) (noisy-or (occurs x t) (strength x a) y 1.0)
false
(parents a))))

The sampling hypothesis
Cortex as hierarchical Bayesian modeler

Barlow, Lee & Mumford, Hinton, Dayan, Zemel, Olshausen, Pouget, Rao, Lewicki, Dean, George & Hawkins, Friston, …

Deep Belief Net

Input Layer of Simple Cells

Figure 1: Graphical Models and their Neural Implementation. (A) Single-level dynamic graphical model. Each circle represents a node denoting the state variable $\theta^t$ which can take on values $\theta_1, \ldots, \theta_N$. (B) Recurrent network for implementing on-line belief propagation for the graphical model in (A). Each circle represents a neuron encoding a state $\theta_t$. Arrows represent synaptic con-
## Computation at COSYNE*09

### Some popular words in titles:
- Feedback: 5
- Circuit: 20
- Gain: 7
- Signal: 5
- Frequency: 8
- Phase: 11
- Correlation: 9
- Nonlinear: 8
- Coding: 12
- Decoding: 13
- Adaptation: 10
- State: 11

### Some less popular words:
- Data structure: 0
- Algorithm: 1
- Symbol: 0
- Pointer: 0
- Buffer: 0
- Graph: 1
- Function: 3
- Language: 0
- Program: 0
- Grammar: 0
- Rule: 1
- Abstract: 1
- Hierarchical: 3
- Recursive: 1
Computation at COSYNE*09

Electrical Engineering
- Feedback: 5
- Circuit: 20
- Gain: 7
- Signal: 5
- Frequency: 8
- Phase: 11
- Correlation: 9
- Nonlinear: 8
- Coding: 12
- Decoding: 13
- Adaptation: 10
- State: 11

Computer Science
- Data structure: 0
- Algorithm: 1
- Symbol: 0
- Pointer: 0
- Buffer: 0
- Graph: 1
- Function: 3
- Language: 0
- Program: 0
- Grammar: 0
- Rule: 1
- Abstract: 1
- Hierarchical: 3
- Recursive: 1