Exploiting compositionality to explore a large space of model structures

Roger Grosse

Dept. of Computer Science, University of Toronto
Introduction

How has the life of a machine learning engineer changed in the past decade?

Many tasks that previously required human experts are starting to be automated.
The probabilistic modeling pipeline

Can we identify good models automatically?

Two challenges:

Automating each stage of this pipeline

Identifying a promising set of candidate models
The probabilistic modeling pipeline

1. Design a model
2. Fit the model
3. Evaluate the model

The process is cyclic, allowing for iterative refinement.
Matrix decompositions

Example: Senate votes, 2009-2010

Votes

Senators

all of one Senator’s votes

record of votes on one motion or bill
Matrix decompositions

Clustering the Senators

Observations  =  Cluster assignments + Cluster centers + Within-cluster variability

Which cluster a Senator belongs to

Which groups of Senators vote for a particular bill/motion
Matrix decompositions

Clustering the Senators

Observations = Cluster assignments + Cluster centers + Within-cluster variability
Matrix decompositions

Clustering the votes

Observations = Cluster centers + Cluster assignments + Within-cluster variability

what sorts of bills/motions one Senator tends to vote for

which Senators tend to vote for one sort of bill/motion

which cluster a vote belongs to
Matrix decompositions

Clustering the votes

Observations = Cluster centers + Cluster assignments + Within-cluster variability
Matrix decompositions

Dimensionality reduction

Observations = Representation of a vote + Residuals

Representation of a Senator
Matrix decompositions

Dimensionality reduction

Observations = \text{something} + \text{Residuals}
Matrix decompositions

Co-clustering Senators and Votes
Matrix decompositions

Co-clustering Senators and Votes
Matrix decompositions

- No structure
- Cluster columns
- Cluster rows
- Dimensionality reduction
- Co-clustering
- ...
The probabilistic modeling pipeline

1. Design a model
2. Fit the model
3. Evaluate the model

Diagram:
- Design a model → Fit the model → Evaluate the model → Design a model
Building models compositionally

We build models by **composing simpler motifs**

- Clustering
- Dimensionality reduction
- Binary attributes
- Heavy-tailed distributions
- Smoothness
- Periodicity
Building models compositionally

(Ghahramani, 1999 NIPS tutorial)
Generative models

**Generation**

Tell a story of how datasets get generated

This gives a joint probability distribution over observations and latent variables

\[ p(h, v) = p(h)p(v|h) \]

**Posterior Inference**

Infer a good explanation of how a particular dataset was generated

Find likely values of the latent variables conditioned on the observations

\[ p(h|v) \]
Space of models: building blocks

<table>
<thead>
<tr>
<th>Gaussian (G)</th>
<th>Multinomial (M)</th>
<th>Integration (C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_i \sim \text{Gamma}(a, b)$</td>
<td>$\pi \sim \text{Dirichlet}(\alpha)$</td>
<td>$u_{ij} = \begin{cases} 1 &amp; \text{if } i \geq j \ 0 &amp; \text{otherwise} \end{cases}$</td>
</tr>
<tr>
<td>$\nu_j \sim \text{Gamma}(a, b)$</td>
<td>$u_i \sim \text{Multinomial}(\pi)$</td>
<td></td>
</tr>
<tr>
<td>$u_{ij} \sim \text{Normal}(0, \lambda_i^{-1}\nu_j^{-1})$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_j \sim \text{Beta}(\alpha, \beta)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$u_{ij} \sim \text{Bernoulli}(p_j)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
We represent models as algebraic expressions.

1. Sample all leaf matrices independently from their corresponding prior distributions

2. Evaluate the resulting expression

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
Space of models: grammar

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
Example: co-clustering

\[ G \rightarrow GM^T + G \]

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
Examples from the literature

\((MG + G)(GM^T + G) + G\)
Bayesian clustered tensor factorization
(Sutskever et al., 2009)

\(\ldots\)

\(M(GM^T + G) + G\)
co-clustering
(e.g. Kemp et al., 2006)

\(\ldots\)

\(MG + G\)
clustering

\((exp(GG + G) \circ G)G + G\)
dependent gaussian scale mixture
(e.g. Karklin and Lewicki, 2005)

\(B(GB^T + G) + G\)
binary matrix factorization
(Meeds et al., 2006)

\((exp(G) \circ G)G + G\)
sparse coding
(e.g. Olshausen and Field, 1996)

\(BG + G\)
binary features
(Griffiths and Ghahramani, 2005)

\(GG + G\)
low-rank approximation
(Salakhutdinov and Mnih, 2008)

\((CG + G)G + G\)
linear dynamical system

\(CG + G\)
random walk

\(G\)
no structure

Grosse, Salakhutdinov, Freeman,
and Tenenbaum, UAI 2012
The probabilistic modeling pipeline

Design a model → Fit a model → Evaluate the model

Posterior Inference
Algorithms: posterior inference

Recursive initialization

implement one algorithm per production rule

share computation between models

Choose the model dimension using Bayesian nonparametrics

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
Posterior inference algorithms

Can make use of **model-specific algorithmic tricks** carefully designed for individual production rules:

- **Eliminating variables analytically**
- **Tractable substructures**
- **Linear algebra identities**
- **High-level transition operators**

\[(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}\]
We evaluate models on the probability they assign to held-out subsets of the observation matrix.
The probabilistic modeling pipeline

Want to search over the large, open-ended space of models

Key problem: the search space is very large!

over 1000 models reachable in 3 productions

how to choose a promising set of models to evaluate?
Algorithms: structure search

A brief history of models of natural images...

Sanger, 1988
Model patches as linear combinations of uncorrelated basis functions
Fourier representation

Olshausen and Field, 1994
Model the heavy-tailed distributions of coefficients
oriented edges similar to simple cells

Model the dependencies between scales of coefficients
high-level texture representation similar to complex cells
Algorithms: structure search

Refining models = applying productions

Based on this intuition, we apply a greedy search procedure

\[ M(GM^T + G) + G \]

\[ MG + G \]

\[ G \]
Experiments: simulated data

Tested on simulated data where we know the correct structure

\[ \sigma^2 = 1 \quad \rightarrow \quad \sigma^2 = 3 \quad \rightarrow \quad \sigma^2 = 10 \]

- low-rank clustering
- binary latent features
- co-clustering
- binary matrix factorization
- BCTF
- sparse coding
- dependent GSM
- random walk
- linear dynamical system

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
**Experiments: simulated data**

Tested on simulated data where we know the correct structure

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>low-rank clustering</td>
<td>$GG + G$</td>
</tr>
<tr>
<td>binary latent features</td>
<td>$BG + G$</td>
</tr>
<tr>
<td>co-clustering</td>
<td>$M(GM^T + G) + G$</td>
</tr>
<tr>
<td>binary matrix factorization</td>
<td>$(BG + G)B^T + G$</td>
</tr>
<tr>
<td>BCTF</td>
<td>$(MG + G)(GM^T + G) + G$</td>
</tr>
<tr>
<td>sparse coding</td>
<td>$(\exp(G) \circ G)G + G$</td>
</tr>
<tr>
<td>dependent GSM</td>
<td>$\exp(G) \circ G)G + G$</td>
</tr>
<tr>
<td>random walk</td>
<td>$CG + G$</td>
</tr>
<tr>
<td>linear dynamical system</td>
<td>$(CG + G)G + G$</td>
</tr>
</tbody>
</table>

Usually chooses the correct structure in low-noise conditions

\[ \sigma^2 = 1 \quad \rightarrow \quad \sigma^2 = 3 \quad \rightarrow \quad \sigma^2 = 10 \]

*Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012*
Experiments: simulated data

Tested on simulated data where we know the correct structure

Usually chooses the correct structure in low-noise conditions

Gracefully falls back to simpler models under heavy noise

---

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma^2 = 1$</th>
<th>$\sigma^2 = 3$</th>
<th>$\sigma^2 = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>low-rank</td>
<td>$GG + G$</td>
<td>$GG + G$</td>
<td>$G$</td>
</tr>
<tr>
<td>clustering</td>
<td>$MG + G$</td>
<td>$MG + G$</td>
<td>$MG + G$</td>
</tr>
<tr>
<td>binary latent features</td>
<td>$BG + G$</td>
<td>$BG + G$</td>
<td>$BG + G$</td>
</tr>
<tr>
<td>co-clustering</td>
<td>$M(GM^T + G) + G$</td>
<td>$M(GM^T + G) + G$</td>
<td>$GM^T + G$</td>
</tr>
<tr>
<td>binary matrix factorization</td>
<td>$(BG + G)B^T + G$</td>
<td>$GG + G$</td>
<td>$GG + G$</td>
</tr>
<tr>
<td>BCTF</td>
<td>$(MG + G)(GM^T + G) + G$</td>
<td>$(exp(G) \circ G)G + G$</td>
<td>$(exp(G) \circ G)G + G$</td>
</tr>
<tr>
<td>sparse coding</td>
<td>$(exp(G) \circ G)G + G$</td>
<td>$(exp(G) \circ G)G + G$</td>
<td>$(exp(G) \circ G)G + G$</td>
</tr>
<tr>
<td>dependent GSM</td>
<td>$(CG + G)G + G$</td>
<td>$(CG + G)G + G$</td>
<td>$(CG + G)G + G$</td>
</tr>
<tr>
<td>random walk</td>
<td>$(CG + G)G + G$</td>
<td>$(CG + G)G + G$</td>
<td>$(CG + G)G + G$</td>
</tr>
<tr>
<td>linear dynamical system</td>
<td>$(CG + G)G + G$</td>
<td>$(CG + G)G + G$</td>
<td>$(CG + G)G + G$</td>
</tr>
</tbody>
</table>

---

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
Experiments: real-world data

Senate votes 09-10

\[ GM^T + G \]

Cluster votes.

22 clusters

largest: party line
Democrat, party line
Republican, all yea
others are series of
votes on single issues

\[ (MG + G)M^T + G \]

Cluster Senators.

11 clusters

no cross-party clusters

—

No third level model improves by more than 1 nat

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
Experiments: real-world data

Senate votes 09-10

Motion capture

Data: motion capture of a person walking. Each row gives a person’s displacement and joint angles in one frame.

\[ GM^T + G \]

\[ CG + G \]

Model 1:
Independent Markov chains

\[ (MG + G)M^T + G \]

\[ C(GG + G) + G \]

Model 2:
Correlations in joint angles

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
Experiments: real-world data

Senate votes 09-10: \(GM^T + G\) (\((MG + G)M^T + G\))

Motion capture: \(CG + G\) (\(C(GG + G') + G\))

Image patches: \(GG + G\) (\((\exp(G') \circ G)G + G\)) (\((\exp(GG + G') \circ G)G + G\))

Data: 1,000 12x12 patches from 10 blurred and whitened images.

Model 1: Low-rank approximation (PCA).

Model 2: Sparsify coefficients to get sparse coding.

Model 3: Model dependencies between scale variables.

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
Experiments: real-world data

Data: Mechanical Turk users’ judgments to 218 questions about 1000 entities

Model 1:
Cluster entities.
39 clusters

Model 2:
Low-rank representation of cluster centers.
8 dimensions
Dimension 1: living vs. nonliving
Dimension 2: large vs. small

Grosse, Salakhutdinov, Freeman, and Tenenbaum, UAI 2012
“Structure discovery in nonparametric regression through compositional kernel search,” ICML 2013.

David Duvenaud, James Lloyd, Roger Grosse, Josh Tenenbaum, and Zoubin Ghahramani,
**Compositional structure search for time series**

Gaussian processes are distributions over functions, specified by kernels.

**Primitive kernels:**
- SE
- PER
- LIN
- RQ

**Composite kernels:**
- LIN × LIN
- SE × PER
- LIN + PER
- LIN × PER

Gaussian processes are distributions over functions, specified by kernels.
Compositional structure search for time series
Compositional structure search for time series

radio critical frequency
An automatic report for the dataset : 01-airline

The Automatic Statistician

Abstract

This report was produced by the Automatic Bayesian Covariance Discovery (ABCD) algorithm.

1 Executive summary

The raw data and full model posterior with extrapolations are shown in figure 1.

![Raw data and full model posterior with extrapolations](image.png)

Figure 1: Raw data (left) and model posterior with extrapolation (right)

The structure search algorithm has identified four additive components in the data. The first 2 additive components explain 98.5% of the variation in the data as shown by the coefficient of determination ($R^2$) values in table 1. The first 3 additive components explain 99.8% of the variation in the data. After the first 3 components the cross validated mean absolute error (MAE) does not decrease by more than 0.1%. This suggests that subsequent terms are modelling very short term trends, uncorrelated noise or are artefacts of the model or search procedure. Short summaries of the additive components are as follows:
10 minute break