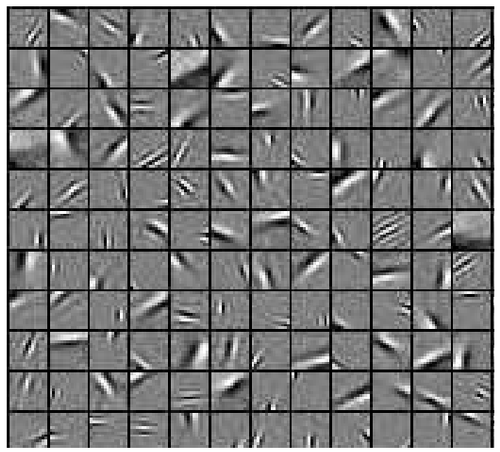
Model selection in a large compositional space

Roger Grosse, Ruslan Salakhutdinov,
Bill Freeman, and Josh Tenenbaum

Motivation

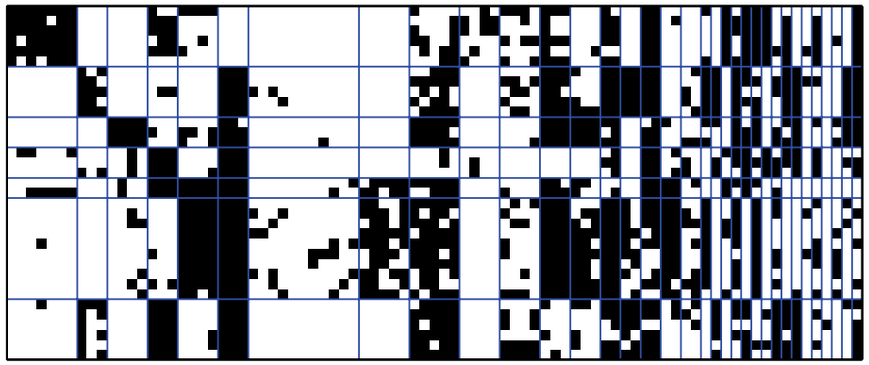
- Recent proliferation of richly structured probabilistic models for matrix data, e.g.

sparse coding



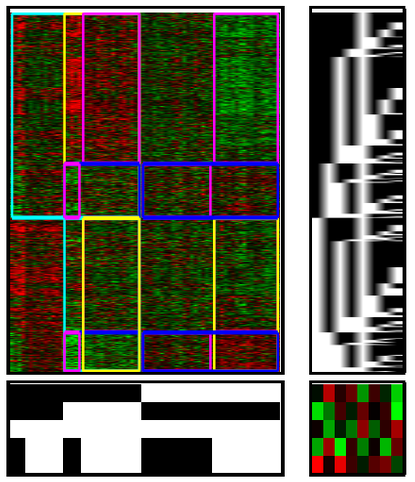
Lee et al. (2006)

infinite relational model
(co-clustering)



Kemp et al. (2006)

binary matrix
factorization



Meeds et al. (2007)

- How do we know which one to use?

Motivation

- How to analyze any dataset:

1. Implement every generative model ever published
2. Fit each of them to the dataset
3. Evaluate all of them using your favorite model selection criterion (e.g. marginal likelihood, description length)
4. Use the one with the highest score

Motivation

- How to analyze any dataset:

  1. Implement every generative model ever published

  2. Fit each of them to the dataset

  3. Evaluate all of them using your favorite model selection criterion (e.g. marginal likelihood, description length)

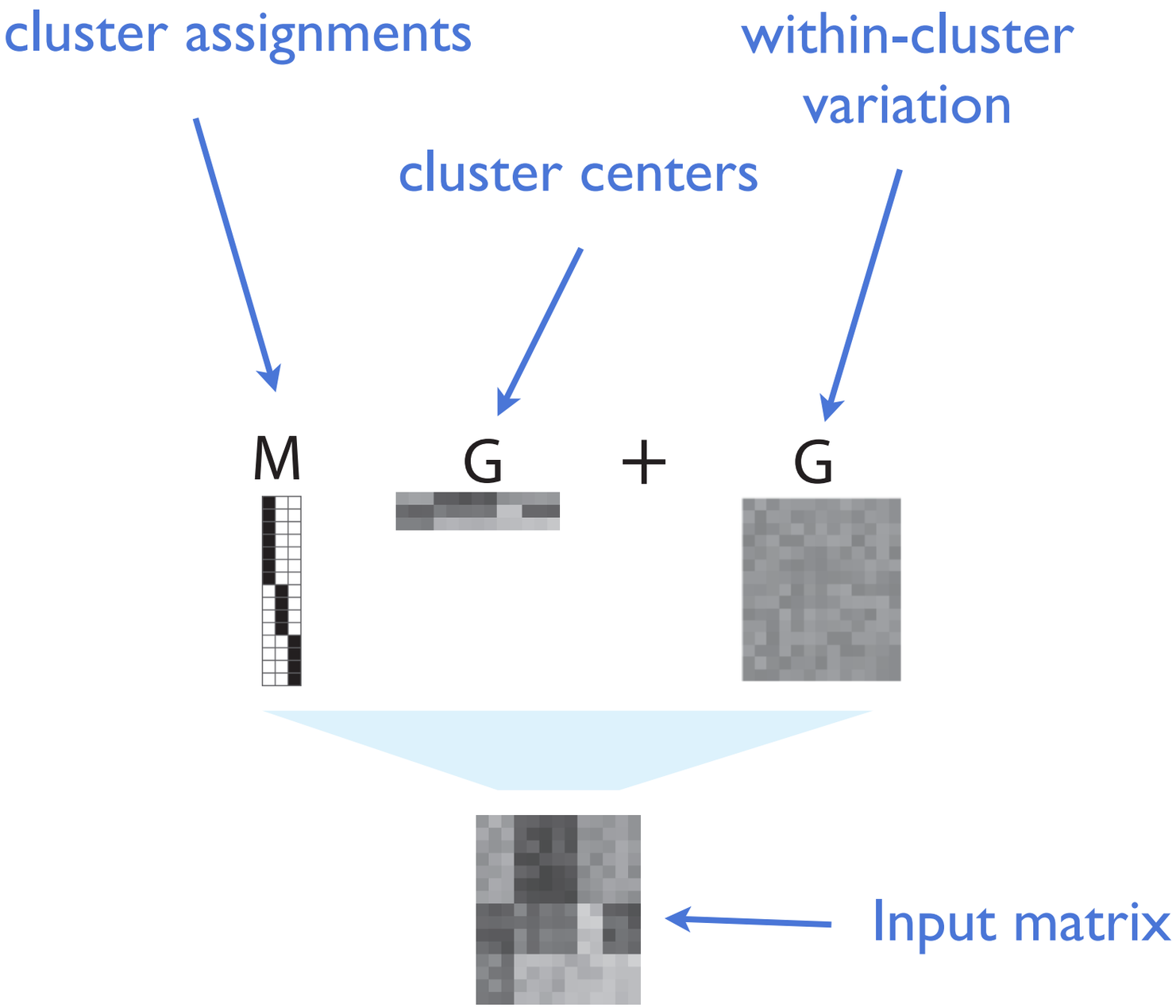
  4. Use the one with the highest score

- What's holding us back?

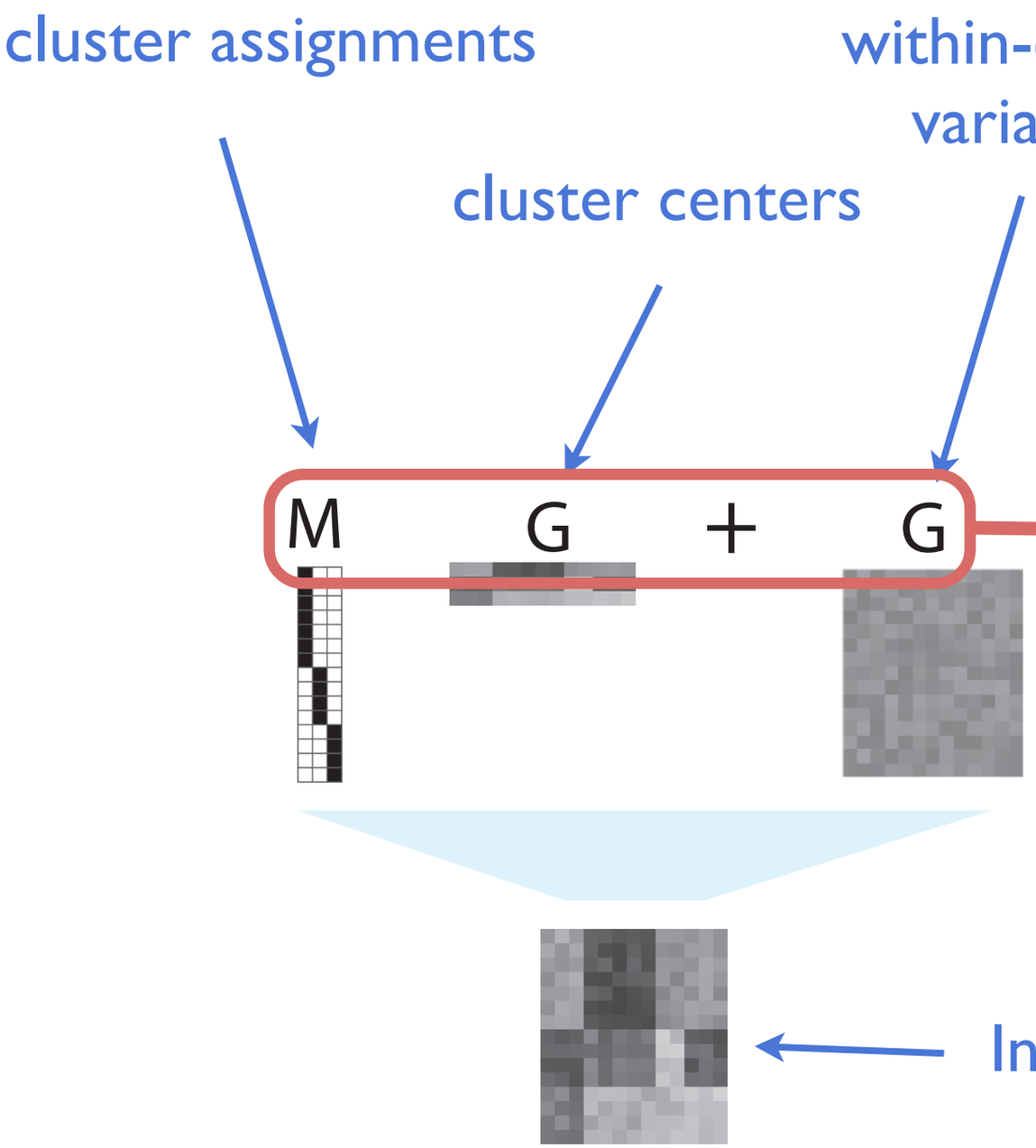
  - implementation

  - computation time

Matrix decompositions



Matrix decompositions

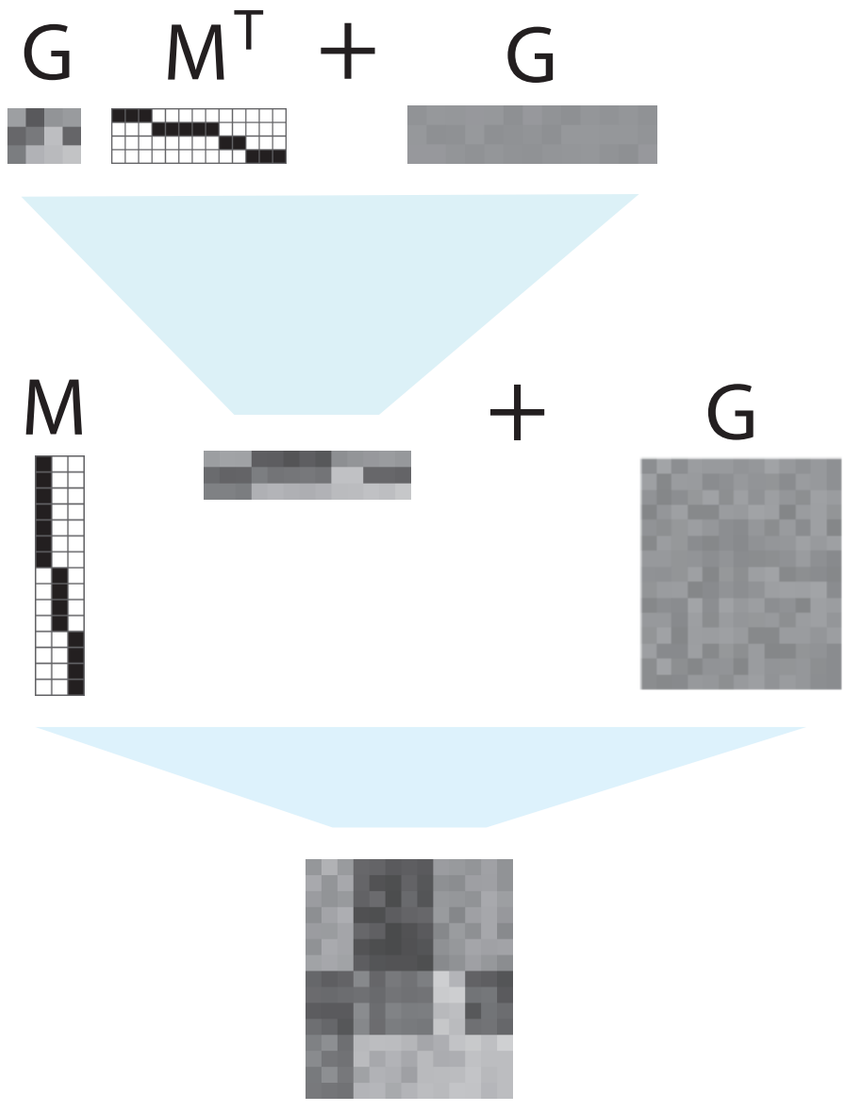


Models are represented as algebraic expressions

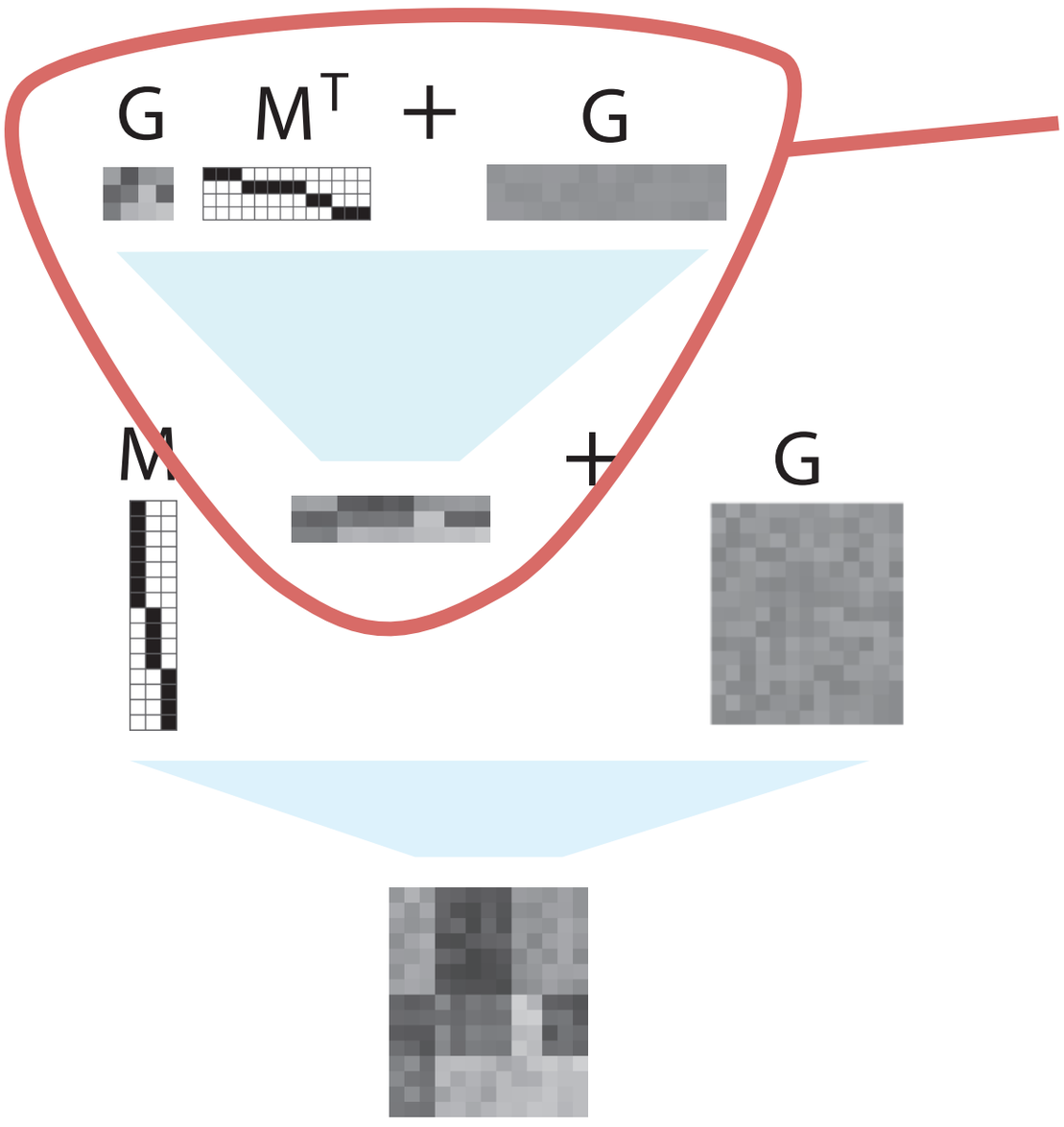
Generate each matrix independently, then apply the arithmetic operations

Input matrix

Matrix decompositions



Matrix decompositions

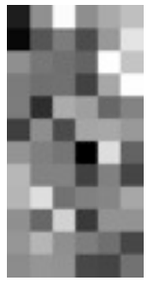


Compositionality

Highly structured models built out of simpler models using well-defined operations

Space of models: building blocks

Space of models: building blocks



Gaussian
(G)

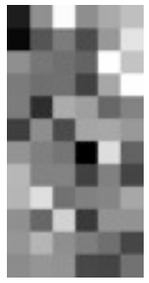
$$\lambda_i \sim \text{Gamma}(a, b)$$

$$\nu_j \sim \text{Gamma}(a, b)$$

$$u_{ij} \sim \text{Normal}(0, \lambda_i^{-1}\nu_j^{-1})*$$

* variance parameters shared between input rows/columns

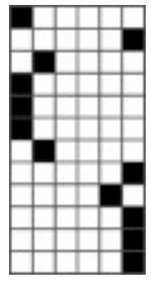
Space of models: building blocks



Gaussian
(G)

$$\lambda_i \sim \text{Gamma}(a, b)$$
$$\nu_j \sim \text{Gamma}(a, b)$$
$$u_{ij} \sim \text{Normal}(0, \lambda_i^{-1}\nu_j^{-1})*$$

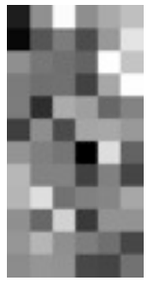
* variance parameters shared between input rows/columns



Multinomial
(M)

$$\pi \sim \text{Dirichlet}(\alpha)$$
$$u_i \sim \text{Multinomial}(\pi)$$

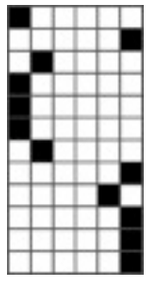
Space of models: building blocks



Gaussian
(G)

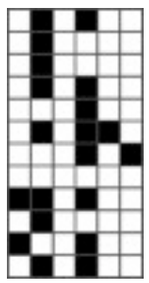
$$\lambda_i \sim \text{Gamma}(a, b)$$
$$\nu_j \sim \text{Gamma}(a, b)$$
$$u_{ij} \sim \text{Normal}(0, \lambda_i^{-1}\nu_j^{-1})*$$

* variance parameters shared between input rows/columns



Multinomial
(M)

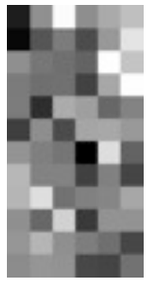
$$\pi \sim \text{Dirichlet}(\alpha)$$
$$u_i \sim \text{Multinomial}(\pi)$$



Bernoulli
(B)

$$p_j \sim \text{Beta}(\alpha, \beta)$$
$$u_{ij} \sim \text{Bernoulli}(p_j)$$

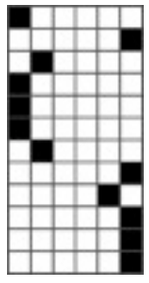
Space of models: building blocks



Gaussian
(G)

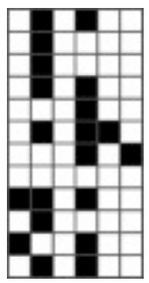
$$\lambda_i \sim \text{Gamma}(a, b)$$
$$\nu_j \sim \text{Gamma}(a, b)$$
$$u_{ij} \sim \text{Normal}(0, \lambda_i^{-1}\nu_j^{-1})*$$

* variance parameters shared between input rows/columns



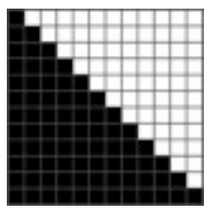
Multinomial
(M)

$$\pi \sim \text{Dirichlet}(\alpha)$$
$$u_i \sim \text{Multinomial}(\pi)$$



Bernoulli
(B)

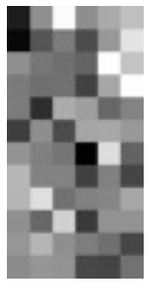
$$p_j \sim \text{Beta}(\alpha, \beta)$$
$$u_{ij} \sim \text{Bernoulli}(p_j)$$



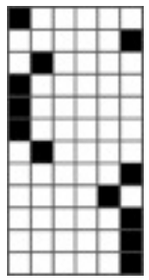
Integration
(C)

$$u_{ij} = \begin{cases} 1 & \text{if } i \geq j \\ 0 & \text{otherwise} \end{cases}$$

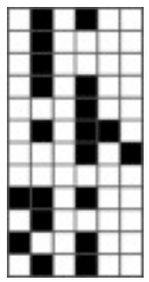
Space of models: grammar



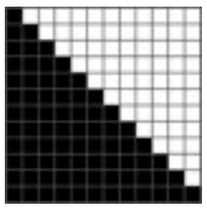
Gaussian
(G)



Multinomial
(M)

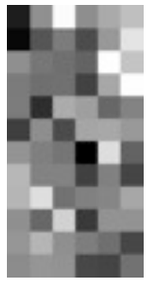


Bernoulli
(B)

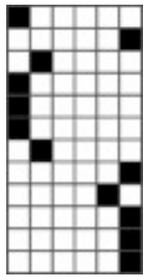


Integration
(C)

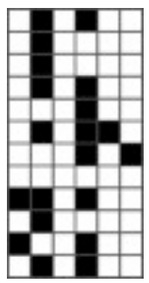
Space of models: grammar



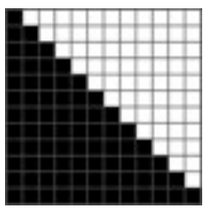
Gaussian
(G)



Multinomial
(M)



Bernoulli
(B)



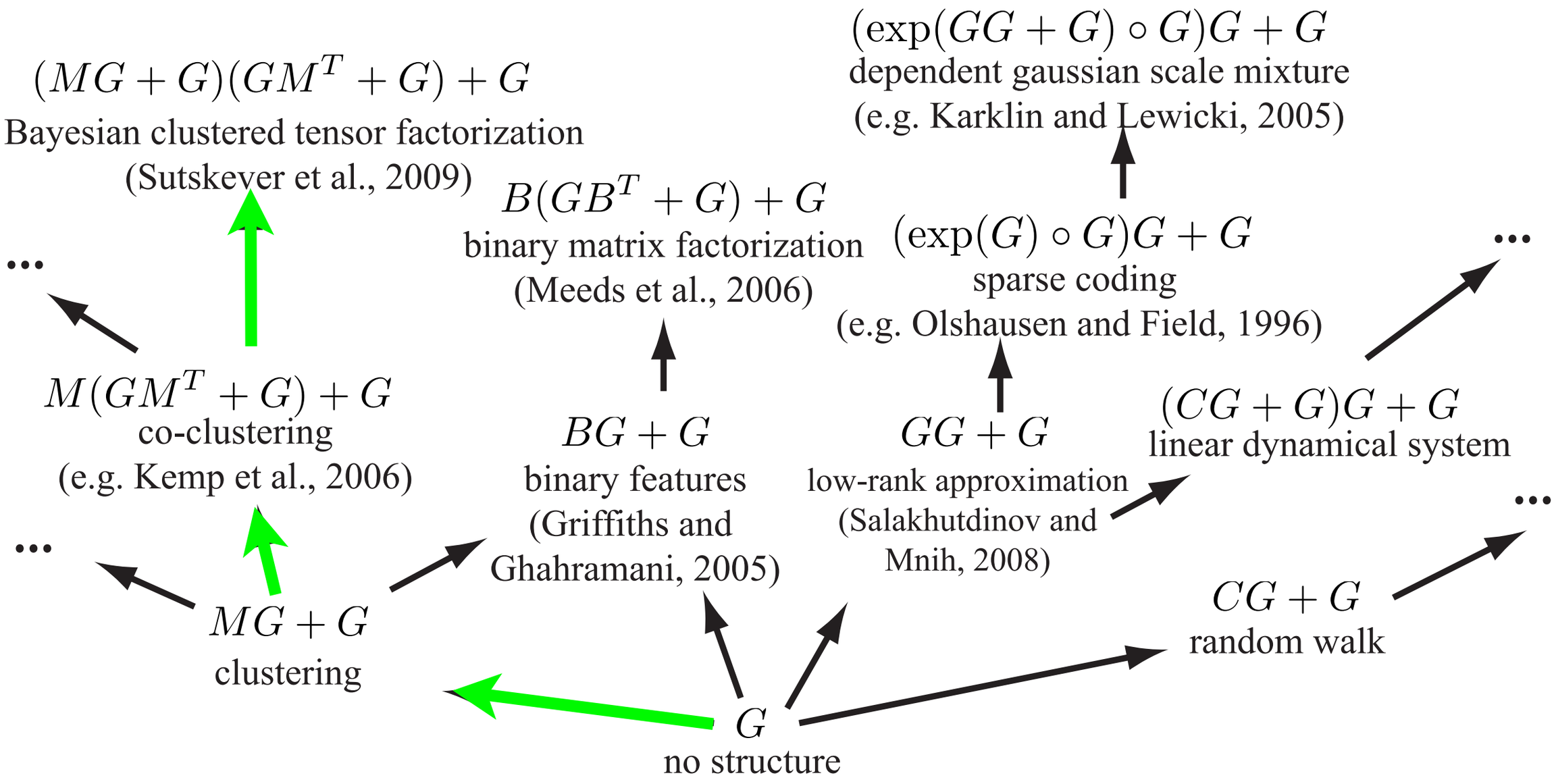
Integration
(C)

Starting symbol: G

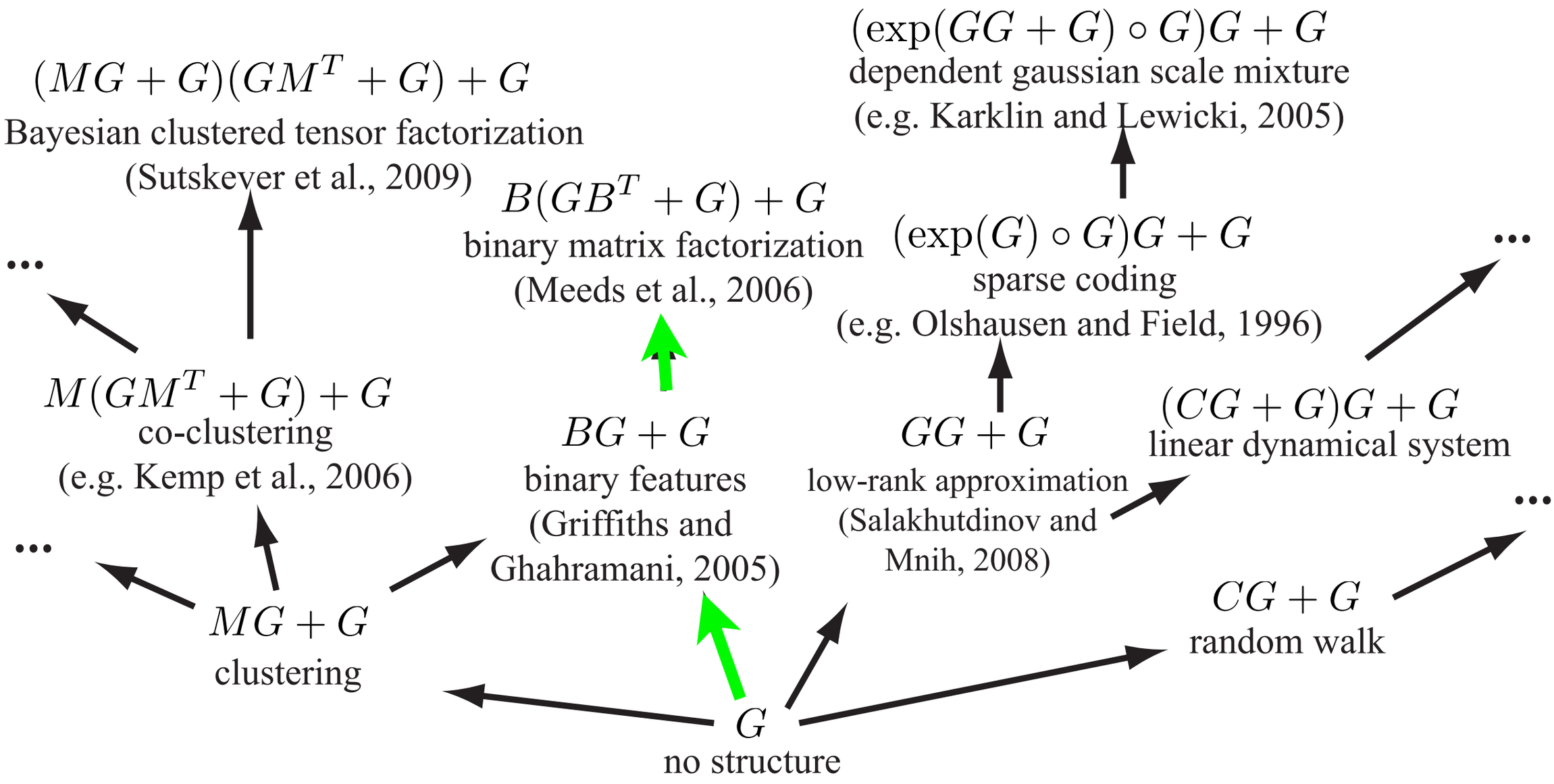
Production rules:

clustering	$G \to MG + G \mid GM^T + G$
	$M \to MG + G$
low rank	$G \to GG + G$
binary features	$G \to BG + G \mid GB^T + G$
	$B \to BG + G$
	$M \to B$
linear dynamics	$G \to CG + G \mid GC^T + G$
sparsity	$G \to \exp(G) \circ G$

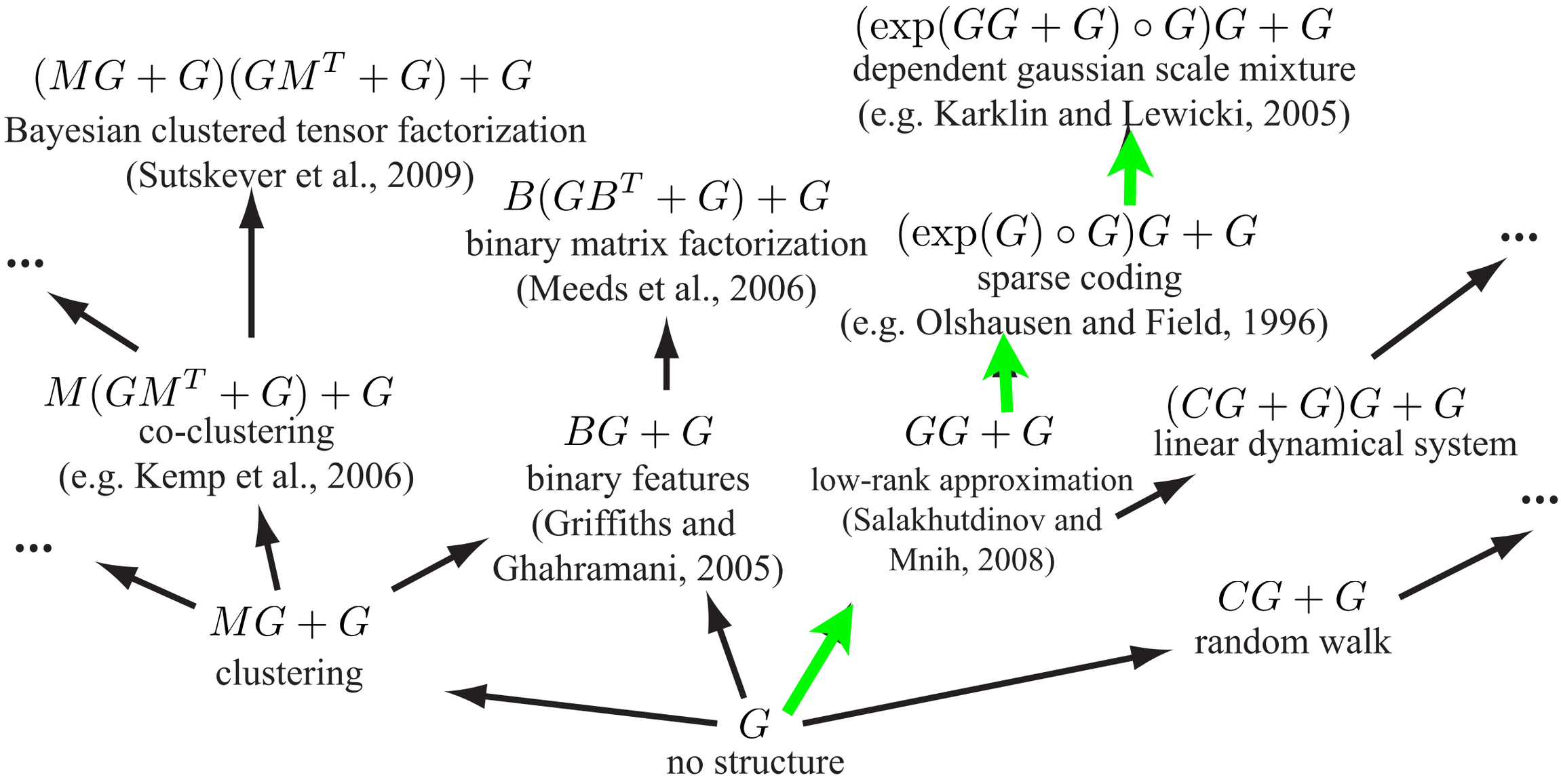
Examples from the literature



Examples from the literature



Examples from the literature

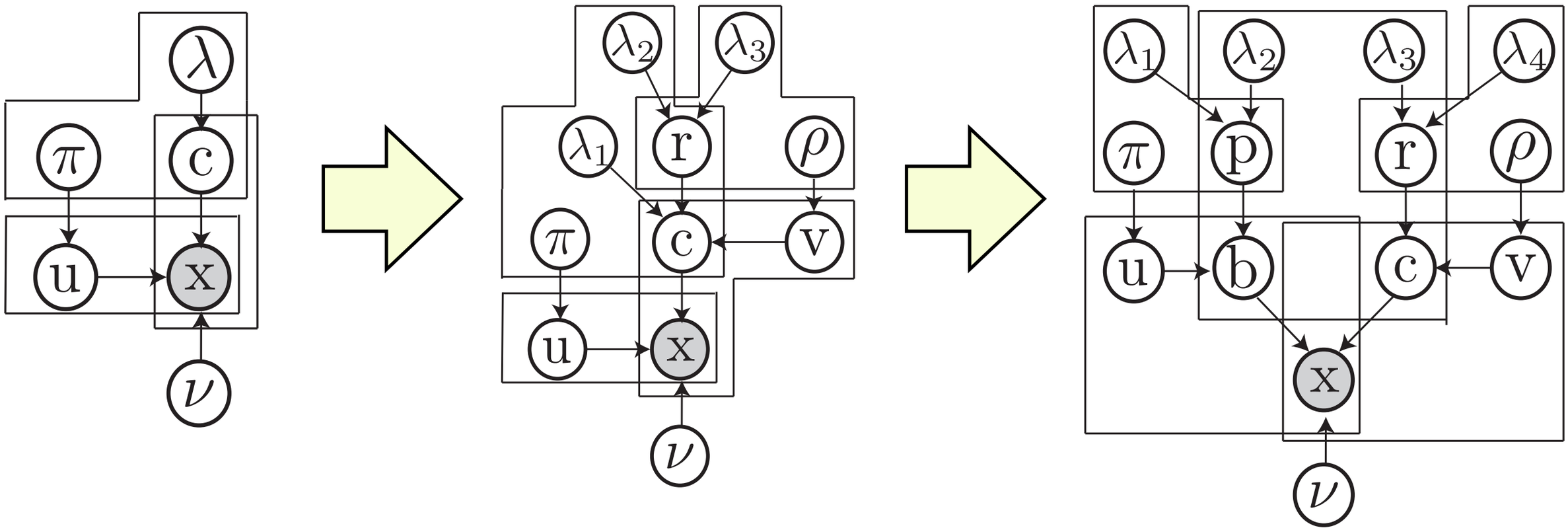


Decompositions as graphical models

$MG + G$

$M(GM^T + G) + G$

$(MG + G)(GM^T + G) + G$



Algebraic expressions give a compact notation for defining a variety of models

Related work

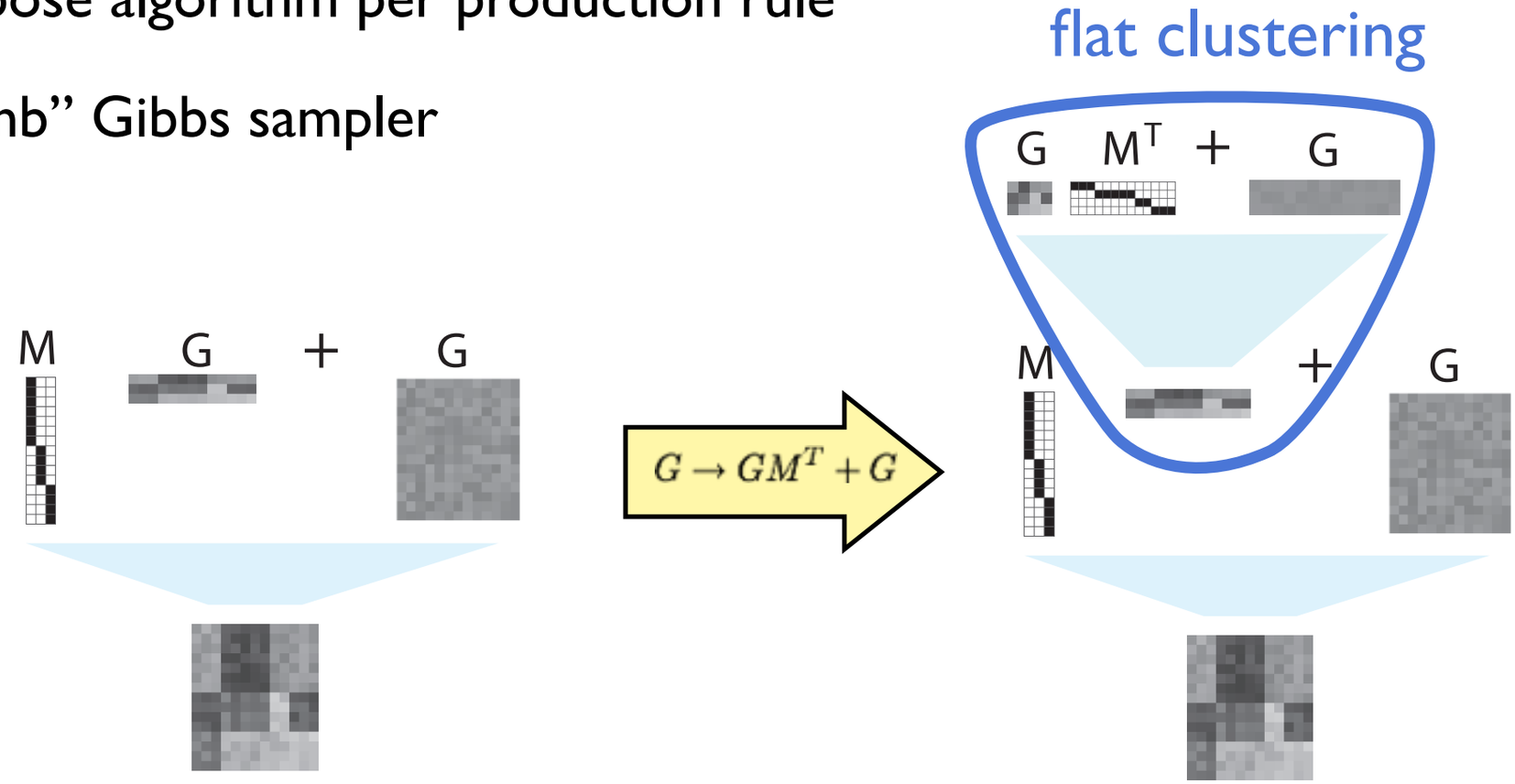
- Algorithmic information theory, e.g. Solomonoff induction (Solomonoff, 1964)
- Structure learning in other domains
  - Bayesian networks (e.g. Teyssier and Koller, 2005)
  - Markov random fields (e.g. Lee et al., 2006)
- Learning the form of graph embeddings (Kemp and Tenenbaum, 2008)
- Equation discovery
  - BACON knowledge discovery engine (Langley, Simon, and Bradshaw, 1984)
  - exploiting context-free grammar (Todorovski and Dzeroski, 1997)
- Matrix factorization frameworks
  - Exponential family PCA (Collins et al., 2002)
  - Roweis and Ghahramani (1999)
  - Singh and Gordon (2008)

Algorithms

- Given a matrix, choose a decomposition model.
- Brute force is impractical
  - nearly 2500 models reachable in 3 production rules
- Need efficient algorithms for:
  - posterior inference in individual models
  - predictive likelihood scoring
  - searching the space of structures

Algorithms: posterior inference

- Inference in these models often requires special-purpose MCMC operators
- Share implementation between models: recursive initialization
  - Each rule is a simple factorization model
  - One special-purpose algorithm per production rule
  - Follow with "dumb" Gibbs sampler



Algorithms: posterior inference

- Smart initialization

Clustering	$G \to MG + G$ $G \to GM^T + G$	Chinese restaurant process, collapsed Gibbs sampling
Binary features	$G \to BG + G$ $G \to GB^T + G$	Indian buffet process, accelerated collapsed Gibbs sampling w/ split merge moves
Low rank	$G \to GG + G$	Probabilistic matrix factorization w/ Poisson prior on # dimensions, reversible jump MCMC
Random walk	$G \to CG + G$ $G \to GC^T + G$	Rauch-Tung-Striebel smoothing

- Choose the latent dimension with Bayesian nonparametrics, then convert to finite model
- Share computation between high-level models

Algorithms: model scoring

Algorithms: model scoring

Entrywise mean squared error

Algorithms: model scoring

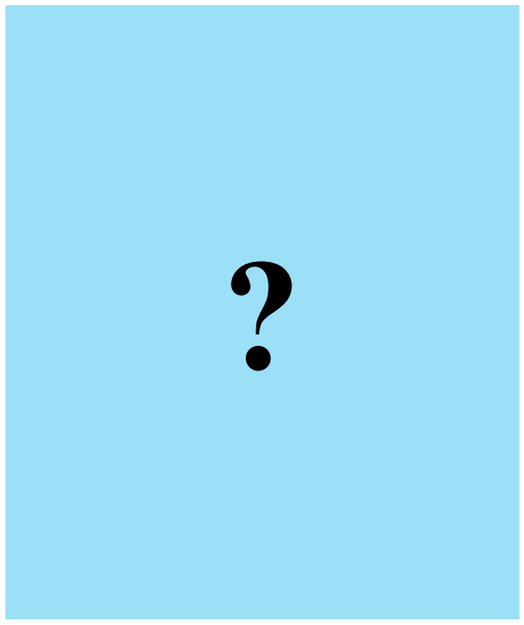
Entrywise mean squared error

Not sensitive enough

Algorithms: model scoring

Entrywise mean squared error

Not sensitive enough

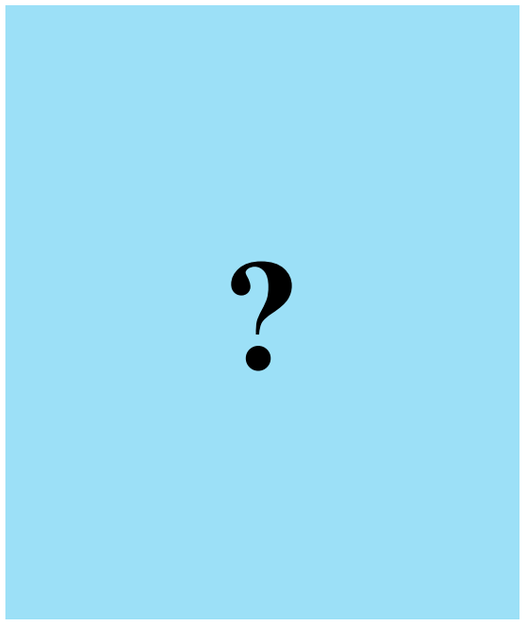


Marginal likelihood

Algorithms: model scoring

Entrywise mean squared error

Not sensitive enough



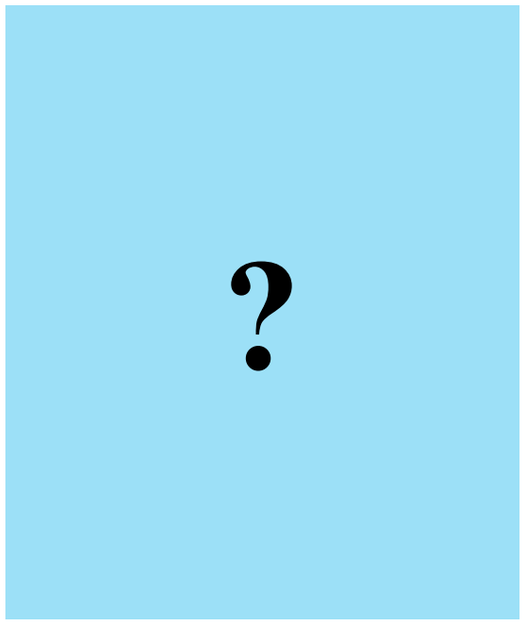
Marginal likelihood

Too hard to compute

Algorithms: model scoring

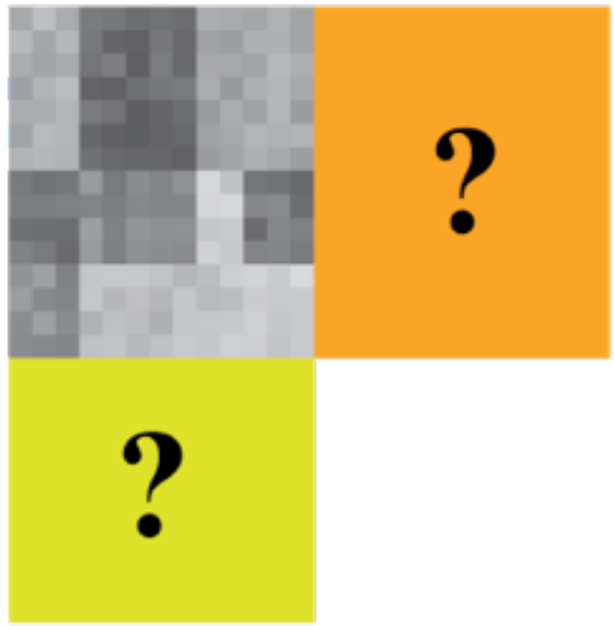
Entrywise mean squared error

Not sensitive enough



Marginal likelihood

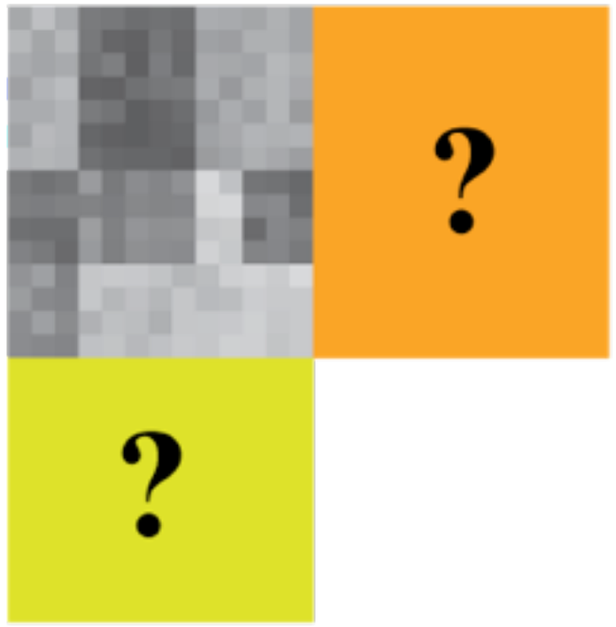
Too hard to compute



Predictive likelihood

Algorithms: predictive likelihood

- Score models using predictive likelihood, the probability of held-out rows and columns conditioned on a training submatrix



- Need a (stochastic) lower bound
- Approximate using a combination of variational and sampling techniques (details in paper)

Algorithms: structure search

- Key problem: choose a promising subset of models to evaluate
- Common research strategy:
  - fit an existing model
  - look for dependencies it doesn't capture
  - add them to the model

Algorithms: structure search

- E.g., this cartoon history of models of image statistics


Bossomaier and Snyder, 1987

Modeling images as linear combinations of uncorrelated basis functions gives a Fourier representation.

Modeling the sparse distribution of the linear reconstruction coefficients gives oriented edges.

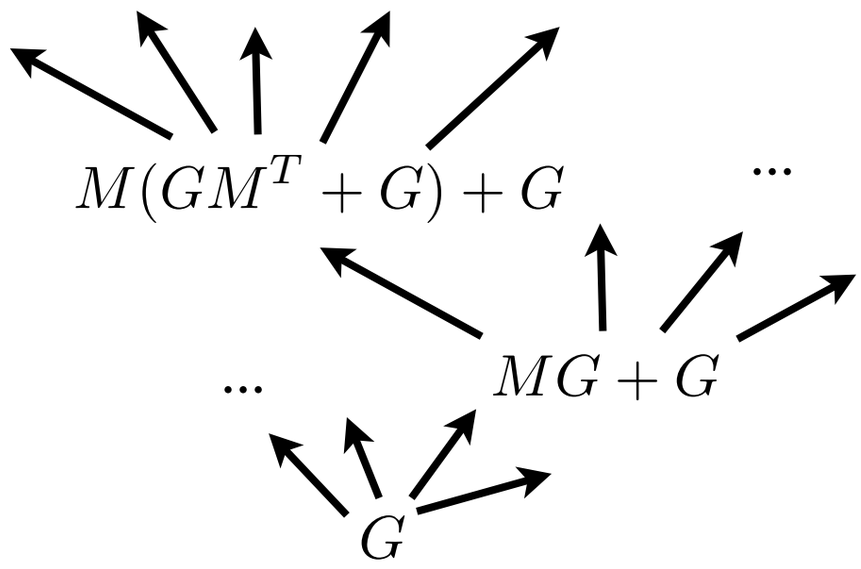

Olshausen and Field, 1996


Karklin and Lewicki, 2005

Modeling the dependencies in the sparsity pattern gives a high-level texture model.

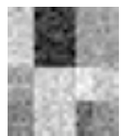
Algorithms: structure search

- Based on this intuition, we apply a greedy search procedure
  - Refining models = applying productions
- Greedy search:
  - expand K best models so far
  - evaluate the resulting models
  - repeat



Experiments: synthetic data

- Tested on synthetic data where we know the structure




— Increasing noise ⟶

$\sigma^2 = 0.1$ $\sigma^2 = 1$ $\sigma^2 = 3$ $\sigma^2 = 10$

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

Experiments: synthetic data

- Tested on synthetic data where we know the structure

— Increasing noise ⟶

	$\sigma^2 = 0.1$	$\sigma^2 = 1$	$\sigma^2 = 3$	$\sigma^2 = 10$
low-rank	$GG + G$	$GG + G$		
clustering	$MG + G$	$MG + G$		
binary latent features	①$(BG + G)G + G$	$BG + G$		
co-clustering	$M(GM^T + G) + G$	$M(GM^T + G) + G$		
binary matrix factorization	①$(BG + G)(GB^T + G) + G$	$(BG + G)B^T + G$		
BCTF	$(MG + G)(GM^T + G) + G$	$(MG + G)(GM^T + G) + G$		
sparse coding	$(\exp(G) \circ G)G + G$	$(\exp(G) \circ G)G + G$		
dependent GSM	①$(\exp(G) \circ G)G + G$	①$(\exp(G) \circ G)G + G$		
random walk	$CG + G$	$CG + G$		
linear dynamical system	$(CG + G)G + G$	$(CG + G)G + G$		

- Usually chooses the right structure in low noise conditions

Experiments: synthetic data

- Tested on synthetic data where we know the structure

— Increasing noise ⟶

	$\sigma^2 = 0.1$	$\sigma^2 = 1$	$\sigma^2 = 3$	$\sigma^2 = 10$
low-rank	$GG + G$	$GG + G$	$GG + G$	①G
clustering	$MG + G$	$MG + G$	$MG + G$	$MG + G$
binary latent features	①$(BG + G)G + G$	$BG + G$	$BG + G$	$BG + G$
co-clustering	$M(GM^T + G) + G$	$M(GM^T + G) + G$	$M(GM^T + G) + G$	①$GM^T + G$
binary matrix factorization	①$(BG + G)(GB^T + G) + G$	$(BG + G)B^T + G$	②$GG + G$	②$GG + G$
BCTF	$(MG + G)(GM^T + G) + G$	$(MG + G)(GM^T + G) + G$	②$GM^T + G$	③G
sparse coding	$(\exp(G) \circ G)G + G$	$(\exp(G) \circ G)G + G$	$(\exp(G) \circ G)G + G$	②G
dependent GSM	①$(\exp(G) \circ G)G + G$	①$(\exp(G) \circ G)G + G$	①$(\exp(G) \circ G)G + G$	③$BG + G$
random walk	$CG + G$	$CG + G$	$CG + G$	①G
linear dynamical system	$(CG + G)G + G$	$(CG + G)G + G$	$(CG + G)G + G$	②$BG + G$

- Usually chooses the right structure in low noise conditions
- Gracefully falls back to simpler models under heavy noise

Experiments: real-world data

Motion capture

Motion capture $CG + G$ $C(GG + G) + G$ —

Data: a person walking in various styles. Each row gives a person's displacement and joint angles in one frame.

Model 1: Independent Markov chains

Model 2: Correlations in joint angles

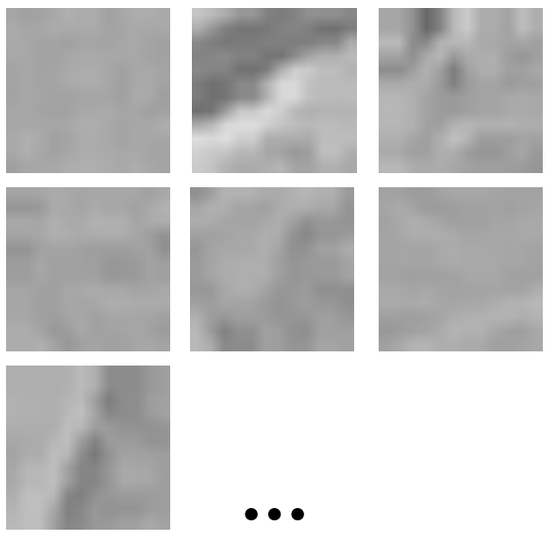
No third level model improves by more than 1 nat

Experiments: real-world data

Image patches

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

Experiments: real-world data

20 Questions

Motion capture	$CG + G$	$C(GG + G) + G$	—
Image patches	$GG + G$	$(\exp(G) \circ G)G + G$	$(\exp(GG + G) \circ G)G + G$
20 Questions	$MG + G$	$M(GG + G) + G$	—

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

Experiments: real-world data

20 Questions

10 largest clusters:

1. **Miscellaneous.** key, chain, powder, aspirin, umbrella, quarter, cord, sunglasses, toothbrush, brush
2. **Clothing.** coat, dress, pants, shirt, skirt, backpack, tshirt, quilt, carpet, pillow, clothing, slipper, uniform
3. **Artificial foods.** pizza, soup, meat, breakfast, stew, lunch, gum, bread, fries, coffee, meatballs, yoke
4. **Machines.** bell, telephone, watch, typewriter, lock, channel, tuba, phone, fan, ipod, flute, aquarium
5. **Natural foods.** carrot, celery, corn, lettuce, artichoke, pickle, walnut, mushroom, beet, acorn
6. **Buildings.** apartment, barn, church, house, chapel, store, library, camp, school, skyscraper
7. **Printed things.** card, notebook, ticket, note, napkin, money, journal, menu, letter, mail, bible
8. **Body parts.** arm, eye, foot, hand, leg, chin, shoulder, lip, teeth, toe, eyebrow, feet, hair, thigh
9. **Containers.** bottle, cup, glass, spoon, pipe, gallon, pan, straw, bin, clipboard, carton, fork
10. **Outdoor places.** trail, island, earth, yard, town, harbour, river, planet, pond, lawn, ocean

- **Our interpretation**, followed by random elements

Experiments: real-world data

Senate votes

Motion capture	$CG + G$	$C(GG + G) + G$	—
Image patches	$GG + G$	$(\exp(G) \circ G)G + G$	$(\exp(GG + G) \circ G)G + G$
20 Questions	$MG + G$	$M(GG + G) + G$	—
Senate votes 09-10	$GM^T + G$	$(MG + G)M^T + G$	—

Cluster votes.

22 clusters

largest: party line Democrat, party line Republican, all yea

others are series of votes on single issues

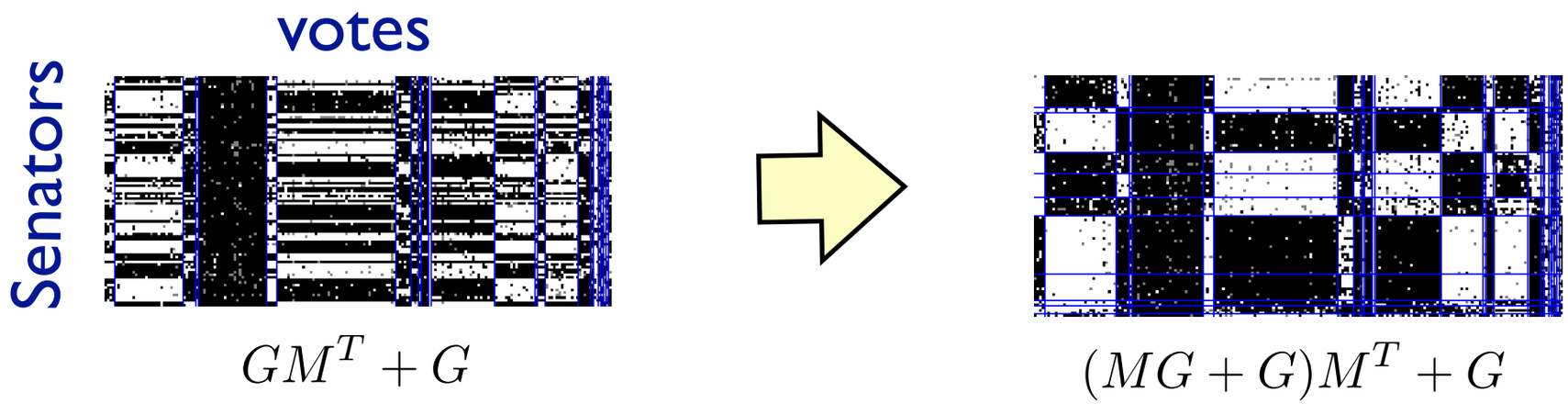
Cluster Senators.

11 clusters

no cross-party clusters

Experiments: real-world data

Senate votes 2009-2010



$GM^T + G$ $(MG + G)M^T + G$

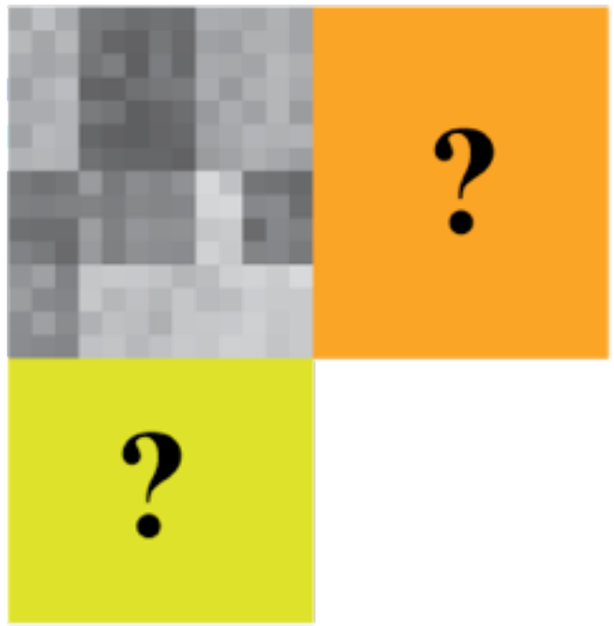
- Rows and columns sorted by cluster, if applicable

Conclusions (of past work)

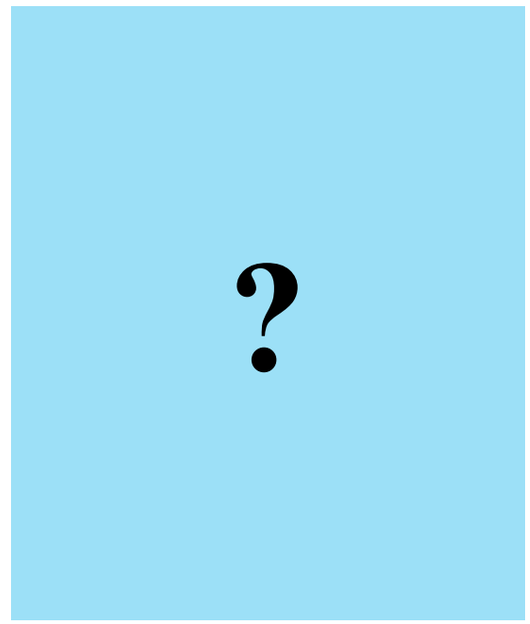
- Compositional framework for matrix decompositions
- Avoiding brute force model selection
  - small toolbox of algorithms corresponding to productions
- Greedy search inspired by the scientific discovery process
  - Low noise: usually finds correct structure
  - High noise: backs off to simple models
  - Learns plausible structures for real-world data
- A step towards automating the discovery of probabilistic models

Current work: marginal likelihood

Entrywise mean squared error



Predictive likelihood



Marginal likelihood

Current work: marginal likelihood

- limitations of predictive likelihood
  - doesn't reward compactness of the learned representation

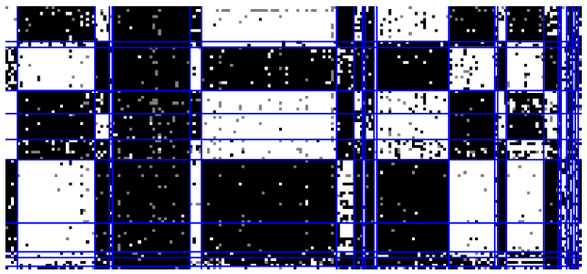
Experiments: real-world data

Senate votes 2009-2010

votes

Senators

$GM^T + G$



$(MG + G)M^T + G$

$(MG + G)(GM^T + G) + G$

The predictive likelihood criterion
didn't choose this production,
even though it would have given
a more compact representation

Current work: marginal likelihood

- limitations of predictive likelihood
  - doesn't reward compactness of the learned representation
  - not appealing as a metaphor for cognition

What if people used predictive likelihood?

Hypothesis 1:
need stimulus spending to jumpstart economy

Hypothesis 2:
cut taxes, reduce regulation, everything will be fine

What if people used predictive likelihood?

Hypothesis 1:
need stimulus spending to jumpstart economy

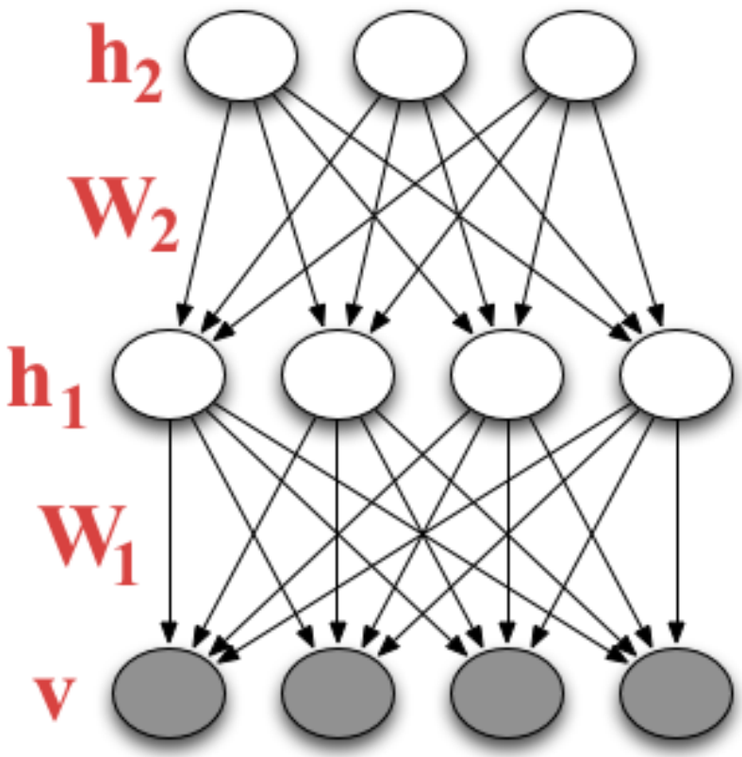
Hypothesis 2:
cut taxes, reduce regulation, everything will be fine

- each hypothesis makes predictions about our future experiences given our past
  - so record a video of the next year of your life, and evaluate the probability pixel-by-pixel under each hypothesis
- really we want to reason with higher level abstractions...

Current work: marginal likelihood

- limitations of predictive likelihood
  - doesn't reward compactness of the learned representation
  - not appealing as a metaphor for cognition
- have to represent the predictive distribution
  - requires thinking about multiple productions at once

Sigmoid belief nets as matrix decompositions



$$h_{1,i}|h_2 \sim \text{Bernoulli}\left(\sigma(w_{2,i}^T h_2 + a_i)\right)$$

$$v_j|h_1 \sim \text{Bernoulli}\left(\sigma(w_{1,j}^T h_1 + b_j)\right)$$

Sigmoid belief nets as matrix decompositions

Need to add a production rule: $B \rightarrow T(BG + G)$

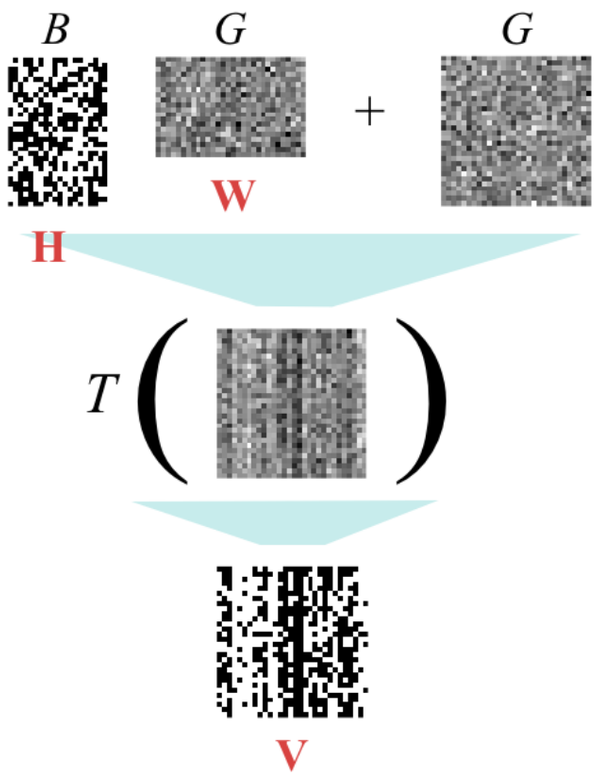
(T is a thresholding operator)

Sigmoid belief nets as matrix decompositions

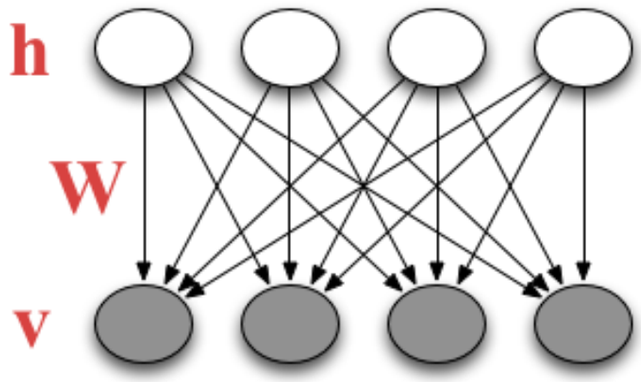
Need to add a production rule: $B \rightarrow T(BG + G)$

(T is a thresholding operator)

Single hidden layer:



matrix decomposition



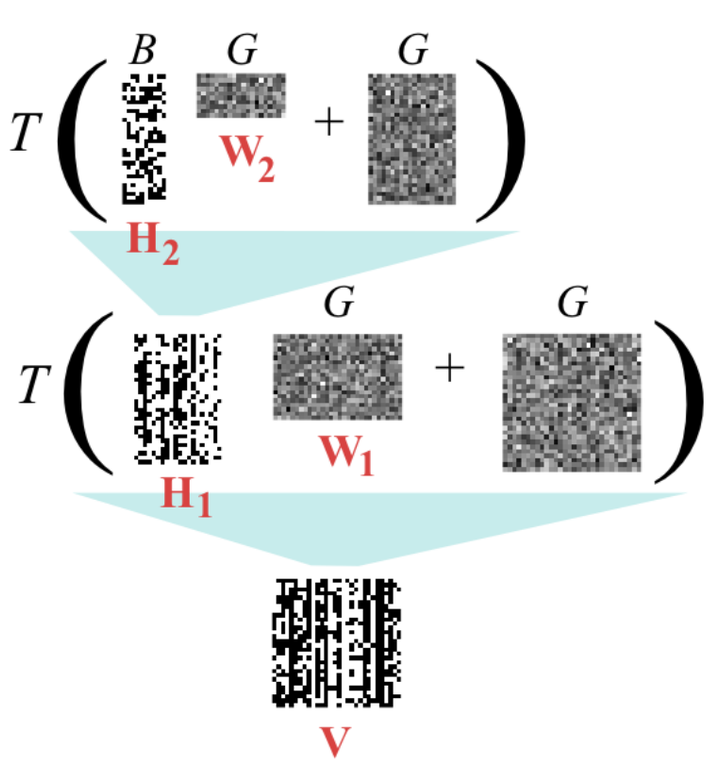
predictive distribution

Sigmoid belief nets as matrix decompositions

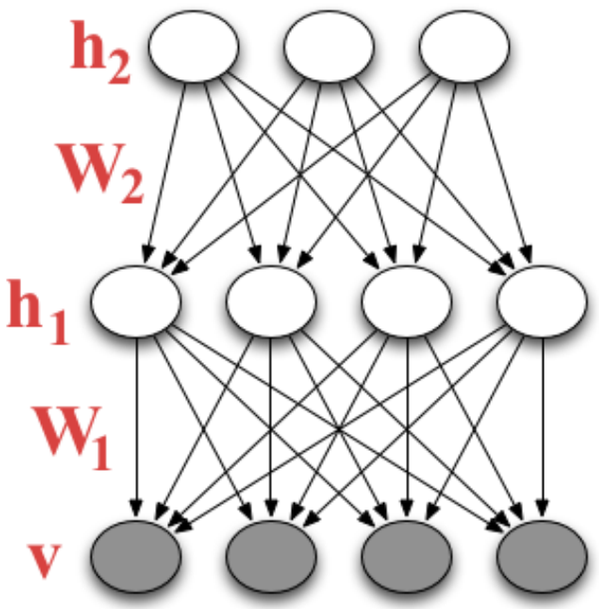
Need to add a production rule: $B \rightarrow T(BG + G)$

(T is a thresholding operator)

Two hidden layers:

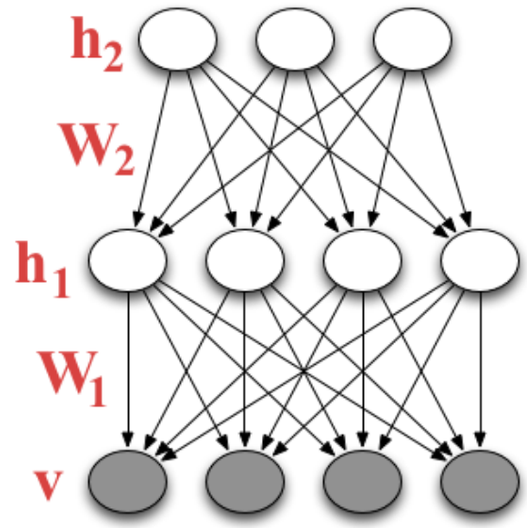


matrix decomposition



predictive distribution

Sigmoid belief nets as matrix decompositions

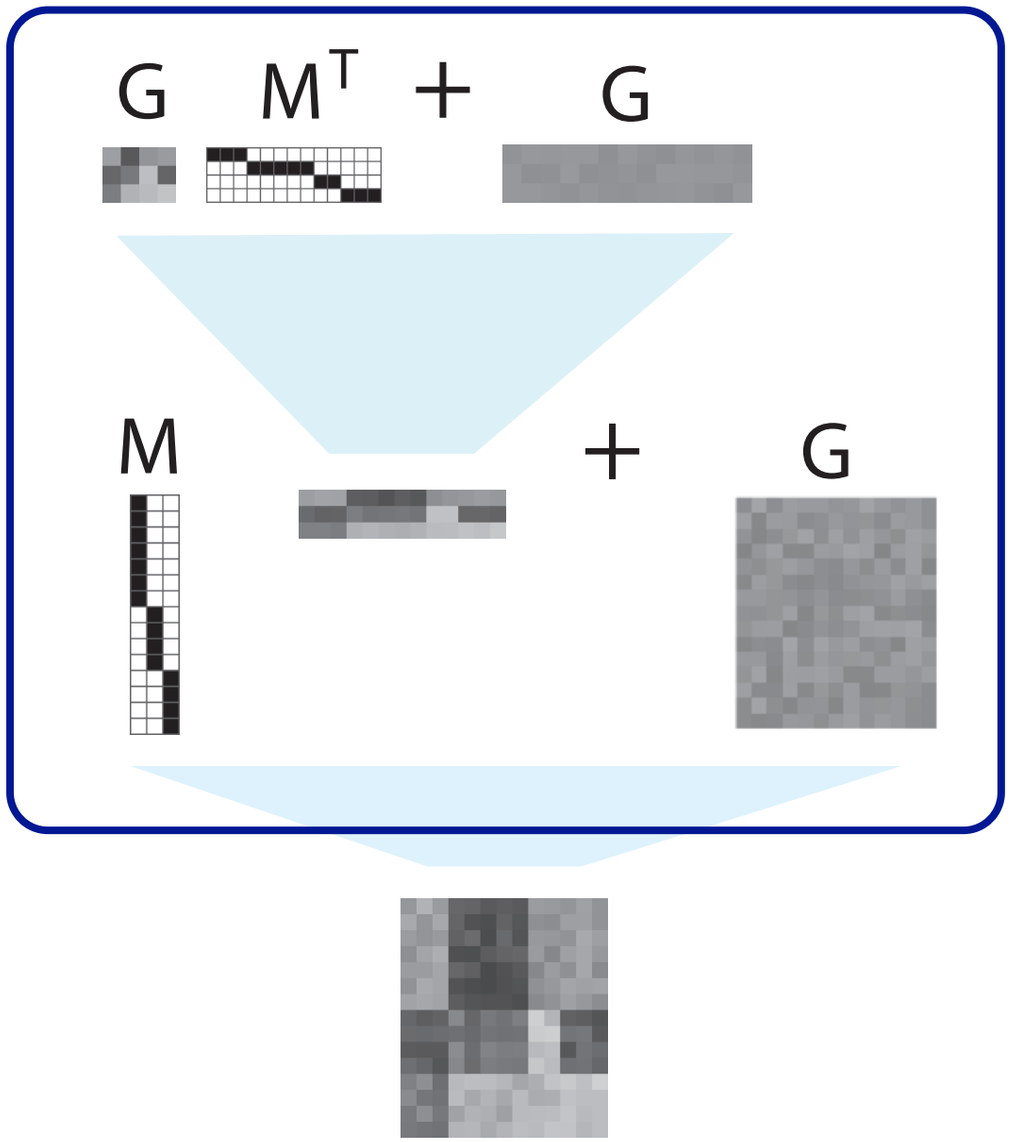


- Why haven't I already implemented this?
  - predictive likelihood: need to integrate out all the hidden units
  - tough inference problem because of explaining away --- one major reason people don't use sigmoid belief nets
- Predictive likelihood computations are the limiting factor when adding new components and observation models

Current work: marginal likelihood

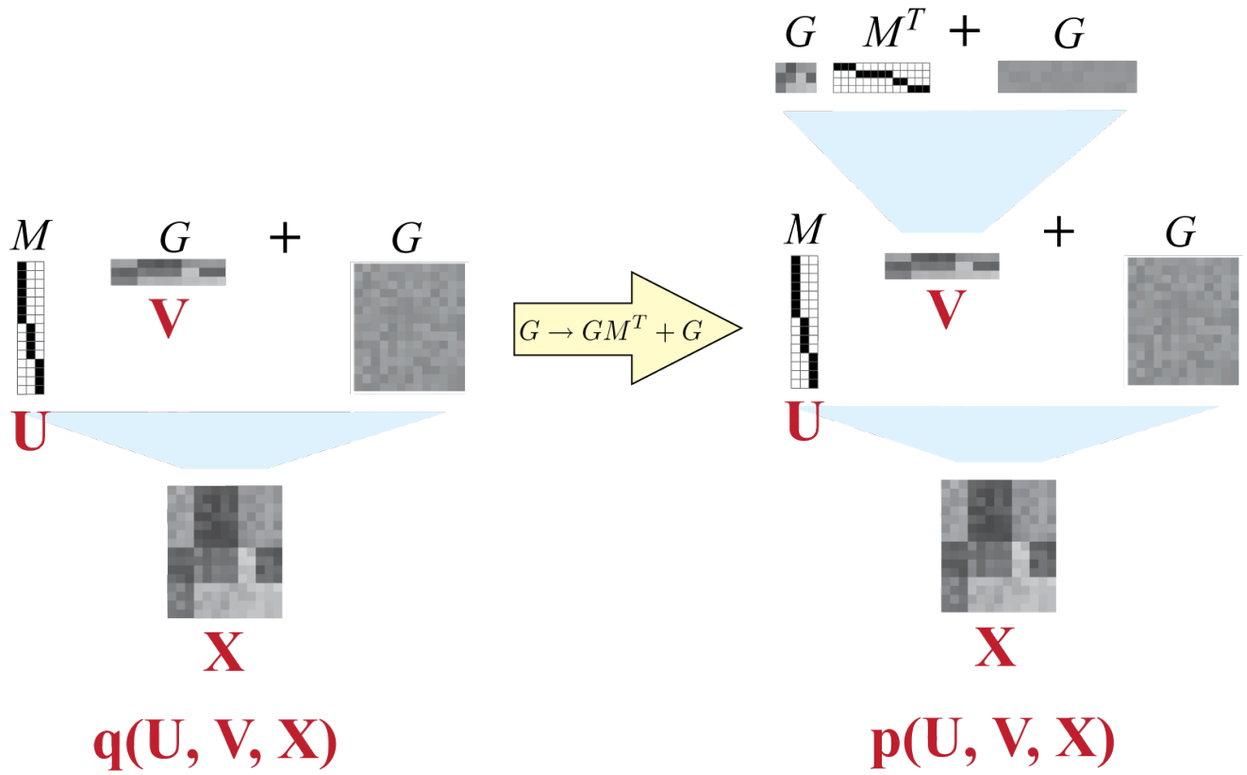
- limitations of predictive likelihood
  - doesn't reward compactness of the learned representation
  - not appealing as a metaphor for cognition
  - have to represent the predictive distribution
    - requires thinking about multiple productions at once
- want to get around these problems by computing marginal likelihood instead

Current work: marginal likelihood

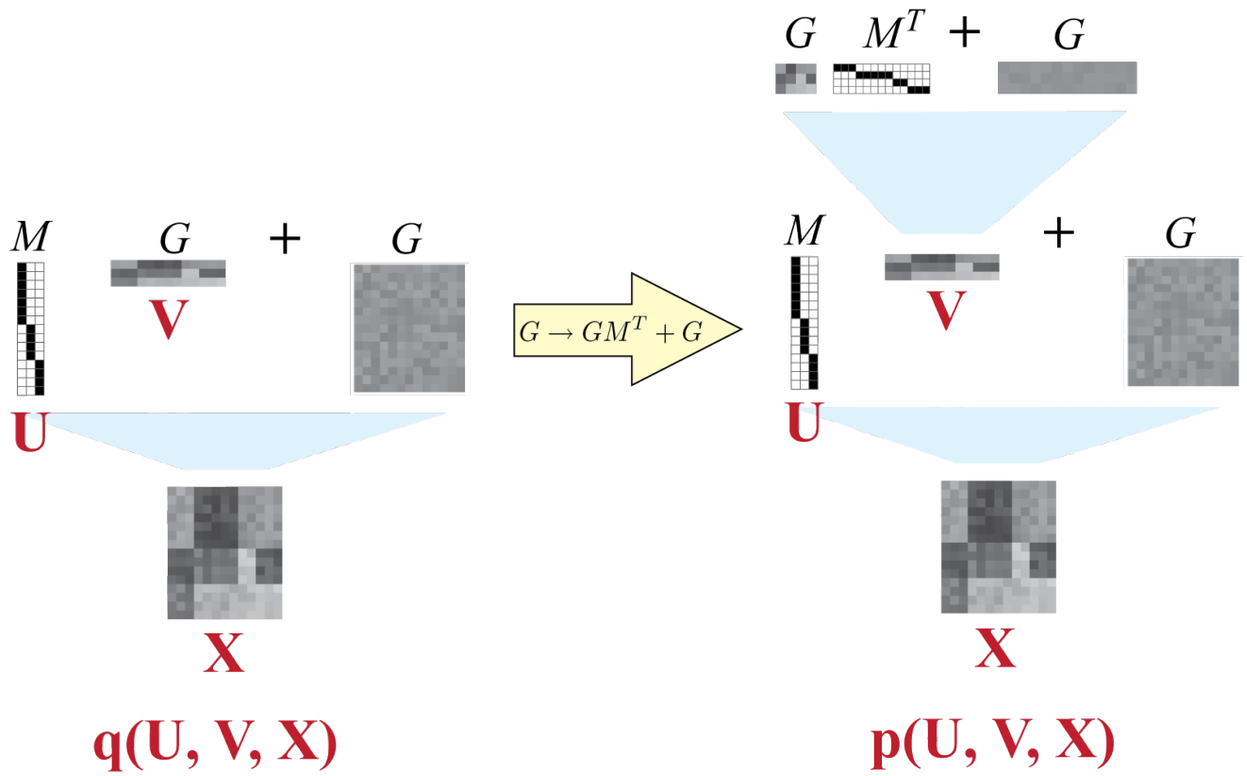


need to integrate out all of the component matrices and their hyperparameters

Compositional importance sampling

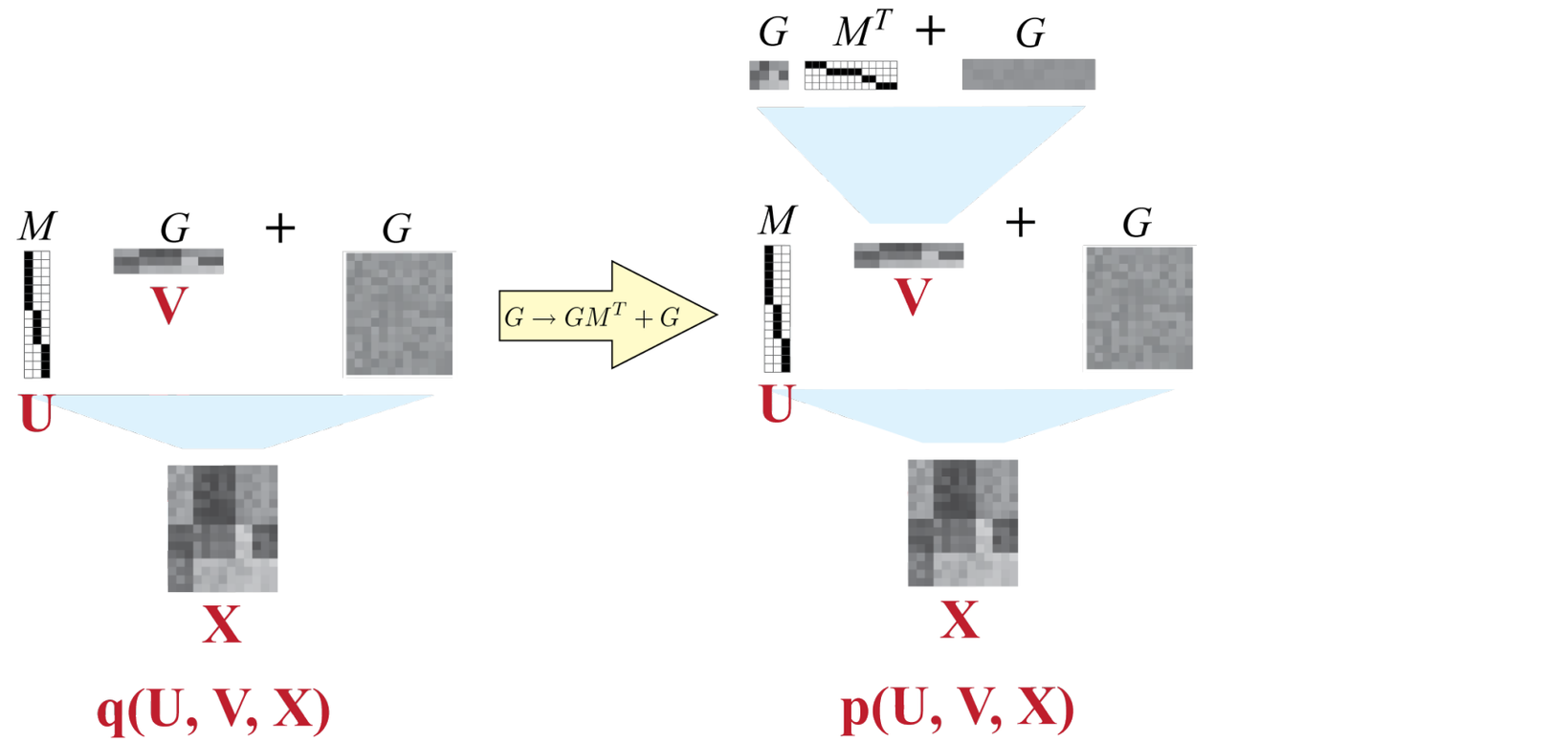


Compositional importance sampling



$$p(X) = q(X)\mathbb{E}_{q(V|X)}\left[\frac{p(V)}{q(V)}\right]$$

Compositional importance sampling



Marginal likelihood under flat clustering

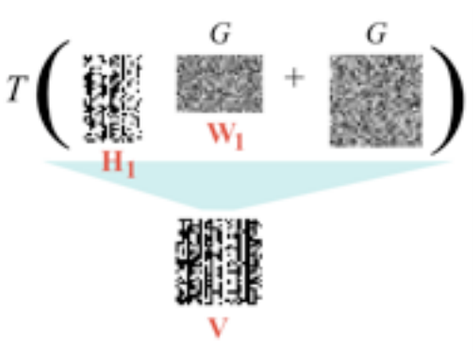
Marginal likelihood under flat clustering

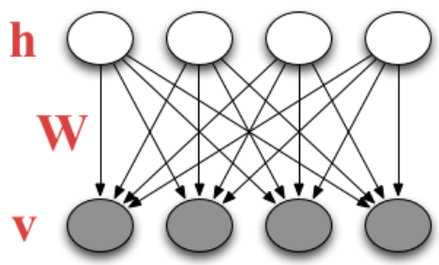
$$p(X) = q(X)\mathbb{E}_{q(V|X)}\left[\frac{p(V)}{q(V)}\right]$$

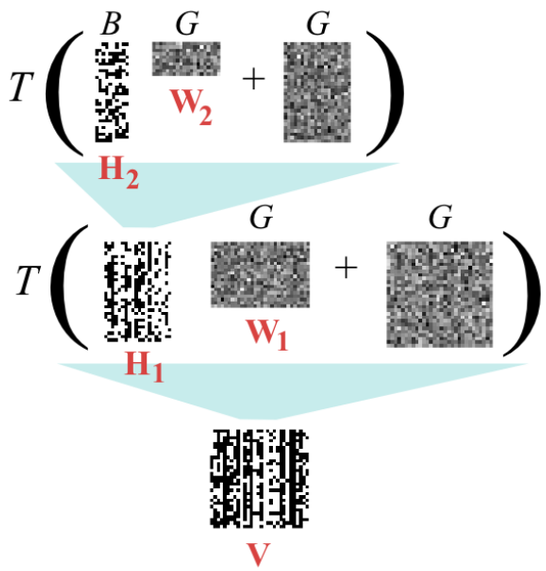
Posterior inference under flat clustering

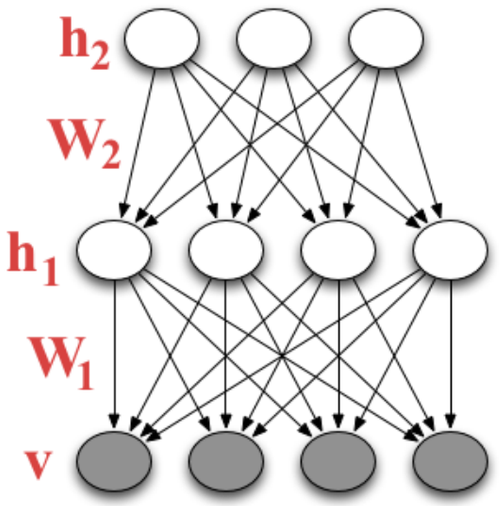
Easy --- Gaussian

Compositional importance sampling







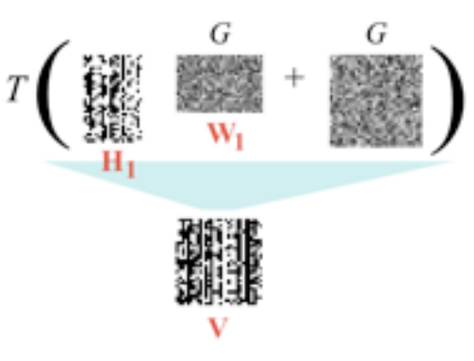


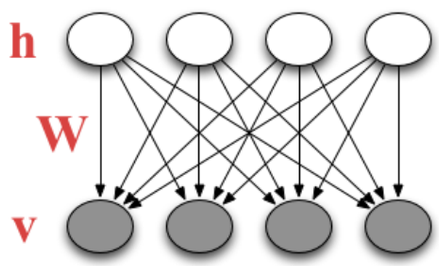
$q(H_1, W_1, V)$

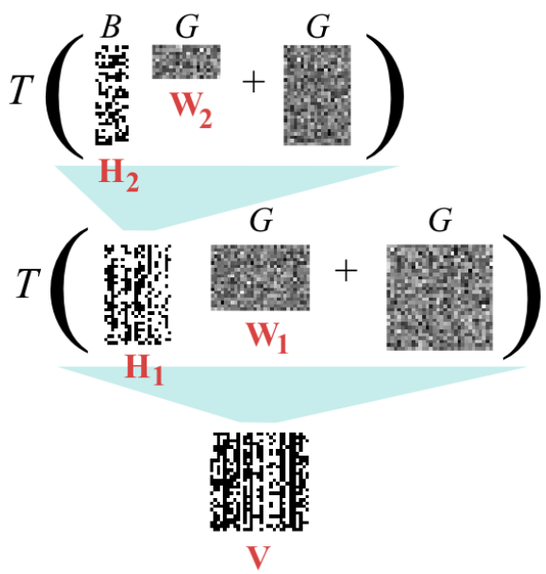
$p(H_1, W_1, V)$

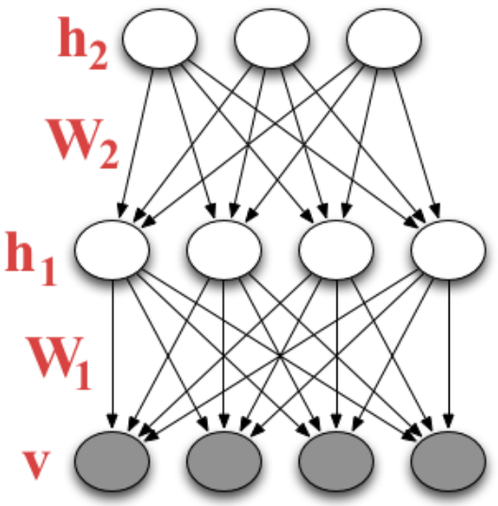
$$p(V) = q(V)\mathbb{E}_{q(H_1|V)}\left[\frac{p(H_1)}{q(H_1)}\right]$$

Compositional importance sampling









$q(H_1, W_1, V)$

$p(H_1, W_1, V)$

$$p'(V) = q(V)\mathbb{E}_{q(H_1|V)}\left[\frac{p(H_1)}{q(H_1)}\right]\mathbb{E}_{p(H_2|H_1)}\left[\frac{p'(H_2)}{p(H_2)}\right]$$

Compositional importance sampling

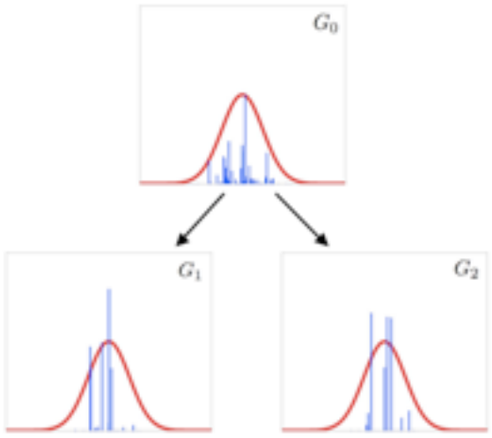
- marginal likelihood estimator for compositional models
- don't keep using the original data; work on higher level abstractions instead
- only need to implement algorithms for posterior sampling and marginal likelihood evaluation for the individual production rules
- hard to implement because marginal likelihood evaluation is hard, even in fairly simple models
- but at least we only need to think about one model at a time

Compositional importance sampling

- Currently working on:
  - implementing marginal likelihood estimators for the production rules (clustering, low rank factorizations, binary factors, etc.)
  - analyzing the bias of the estimator
    - ultimately want a recipe for designing a model class that can be searched over using CIS

Other compositional models

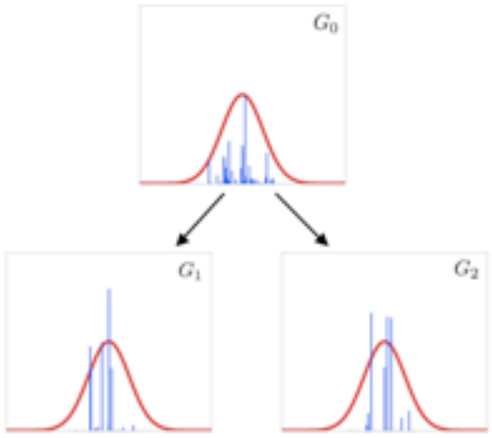
Other compositional models



hierarchical Dirichlet process

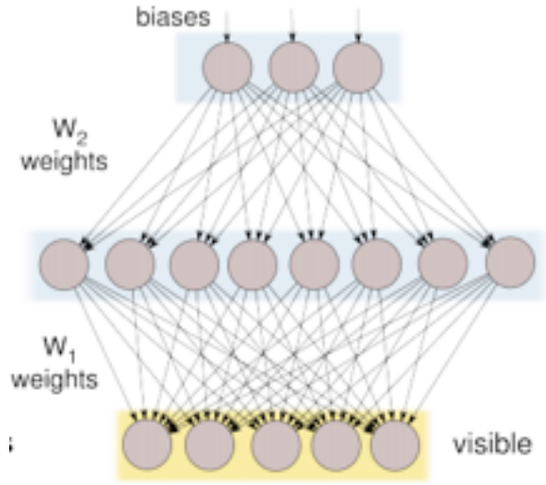
(Blei et al., 2004)

Other compositional models



hierarchical Dirichlet process

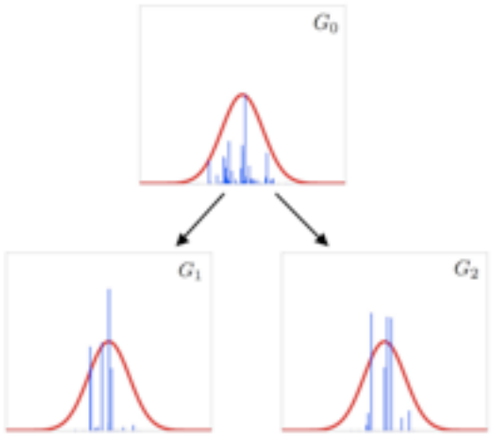
(Blei et al., 2004)



sigmoid belief network

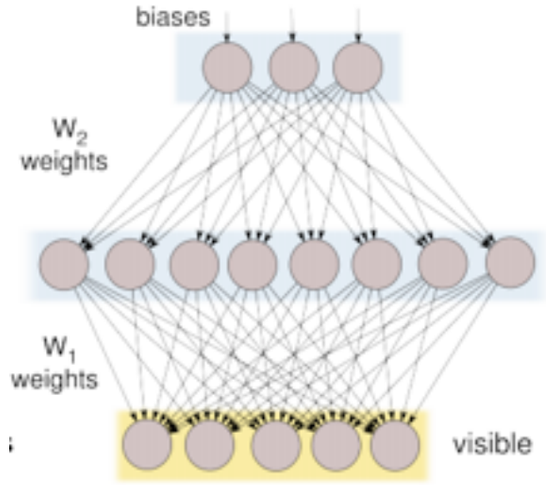
(Neal, 1991)

Other compositional models



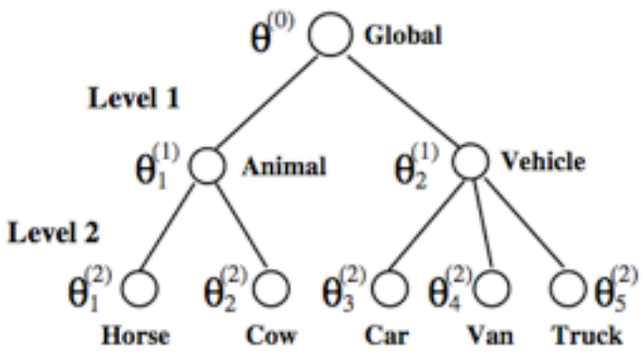
hierarchical Dirichlet process

(Blei et al., 2004)



sigmoid belief network

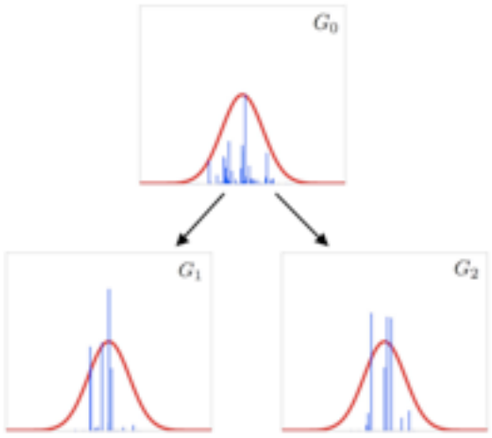
(Neal, 1991)



hierarchical classification

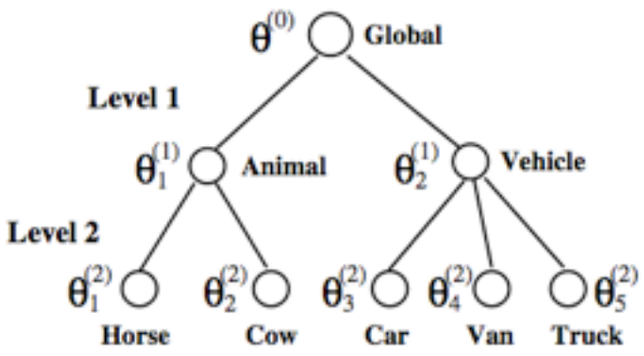
(Salakhutdinov et al., 2011)

Other compositional models



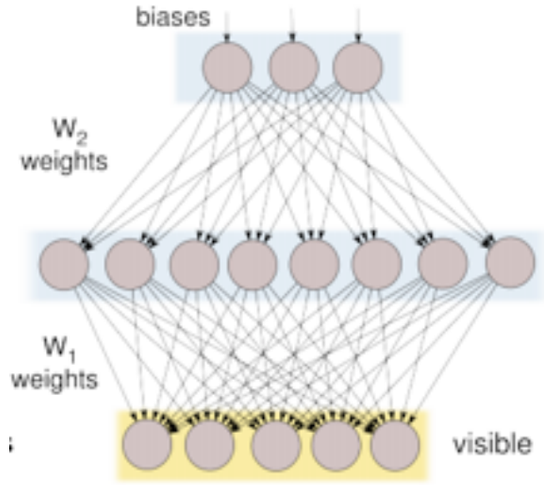
hierarchical Dirichlet process

(Blei et al., 2004)



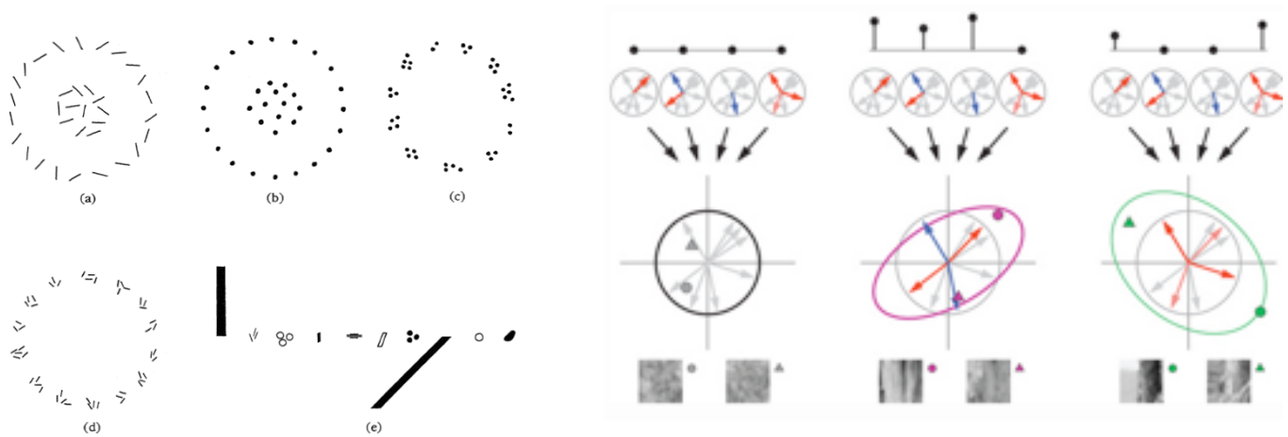
hierarchical classification

(Salakhutdinov et al., 2011)



sigmoid belief network

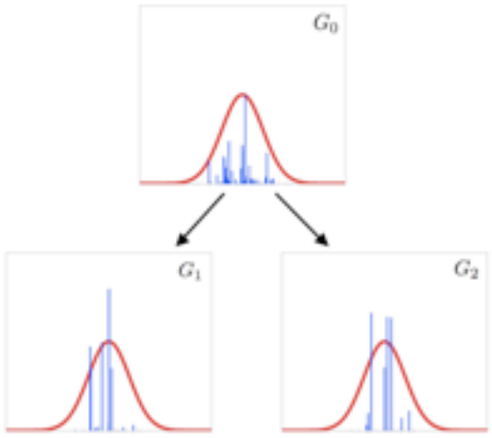
(Neal, 1991)



hierarchical image models

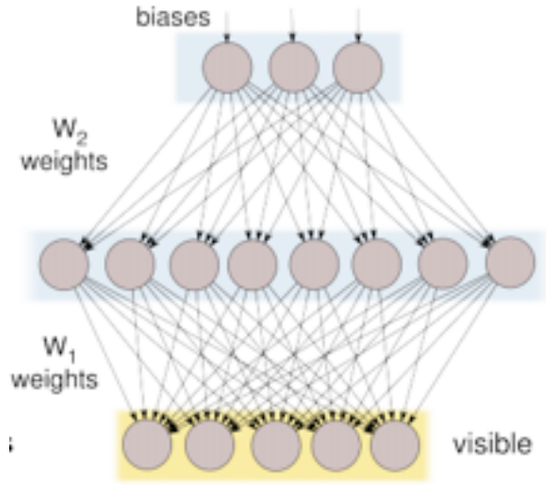
(Marr, 1980; Karklin and Lewicki, 2008)

Other compositional models



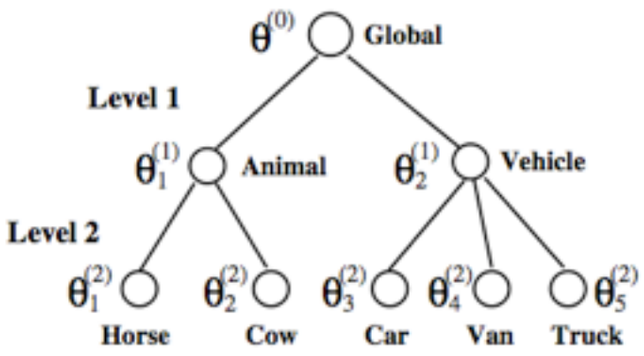
hierarchical Dirichlet process

(Blei et al., 2004)



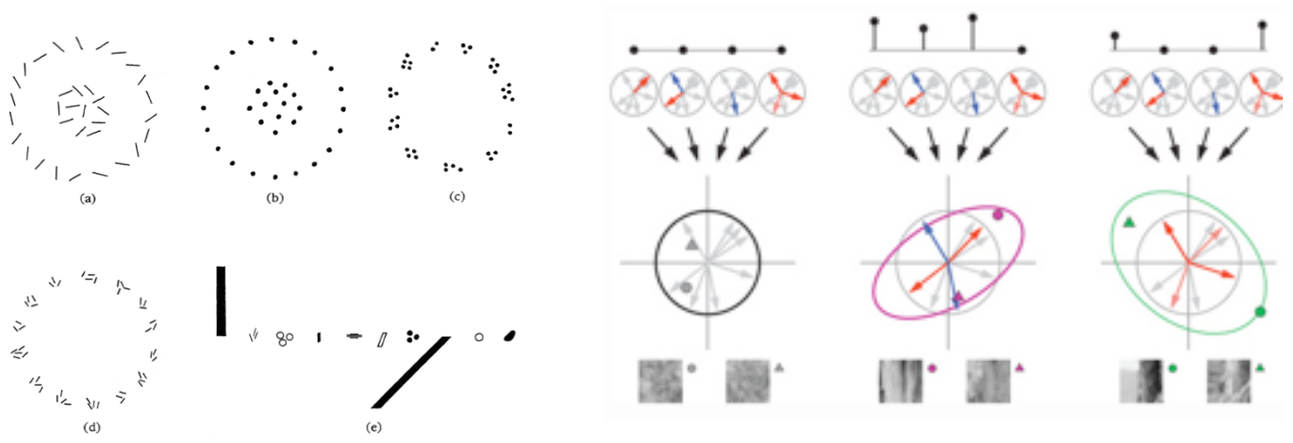
sigmoid belief network

(Neal, 1991)



hierarchical classification

(Salakhutdinov et al., 2011)



hierarchical image models

(Marr, 1980; Karklin and Lewicki, 2008)

can these strategies be applied more generally?