Exploiting compositionality to explore a large space of model structures

R. Grosse, R. Salakhutdinov, W. Freeman, & J. Tenenbaum

Best Student Paper at UAI 2012

Jan Gasthaus

Tea talk
31st Aug 2012

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set
- Ideal approach
  - Implement all models ever published

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set
- Ideal approach
  - Implement all models ever published
  - Fit them to the data set

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set
- Ideal approach
  - Implement all models ever published
  - Fit them to the data set
  - Compare them using some model selection criterion and pick the best

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set
- Ideal approach
  - Implement all models ever published
  - Fit them to the data set
  - Compare them using some model selection criterion and pick the best
- Mainly a computational problem; Proposed solution:

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set
- Ideal approach
  - Implement all models ever published
  - Fit them to the data set
  - Compare them using some model selection criterion and pick the best
- Mainly a computational problem; Proposed solution:
  - Pick a rich class of models: matrix decomposition models

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set
- Ideal approach
  - Implement all models ever published
  - Fit them to the data set
  - Compare them using some model selection criterion and pick the best
- Mainly a computational problem; Proposed solution:
  - Pick a rich class of models: matrix decomposition models
  - Fit more complex models re-using computations from simple ones

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set
- Ideal approach
  - Implement all models ever published
  - Fit them to the data set
  - Compare them using some model selection criterion and pick the best
- Mainly a computational problem; Proposed solution:
  - Pick a rich class of models: matrix decomposition models
  - Fit more complex models re-using computations from simple ones
  - Approximate model selection criterion

Motivation

- **Goal:** Given a data set, determine the right model to use for that data set
- Ideal approach
  - Implement all models ever published
  - Fit them to the data set
  - Compare them using some model selection criterion and pick the best
- Mainly a computational problem; Proposed solution:
  - Pick a rich class of models: matrix decomposition models
  - Fit more complex models re-using computations from simple ones
  - Approximate model selection criterion
  - Greedy heuristic for exploring the space of structure exploiting compositionality

In A Nutshell

- Grammar for generative models for matrix factorization
  - Express models as algebraic expressions such as $MG + G$
  - Devise CFG that generates these expressions with rules like $G \rightarrow GG + G$
- Search over model structures greedily by applying the production rules and using an approximate lower bound on model score
- Initialize sampling in model by using a specialized algorithm for each production rule

Components

1. **Gaussian (G).** Entries are independent Gaussians:

$$u_{ij} \sim \text{Gaussian}(0, \lambda_i^{-1}\lambda_j^{-1}).$$

   This is our most generic component prior, and gives a way of deferring or ignoring structure.[1]

2. **Multinomial (M).** Rows are independent multinomials, with one 1 and the rest 0's:

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

   This is useful for clustering models, where u_i determines the cluster assignment for the i^{th} row.

3. **Bernoulli (B).** Entries are independent Bernoullis:

$$\pi_j \sim \text{Beta}(a, b) \qquad u_{ij} \sim \text{Bernoulli}(\pi_j).$$

   This is useful for binary latent feature models.

4. **Integration matrix (C).** Entries below the diagonal are deterministically 1:

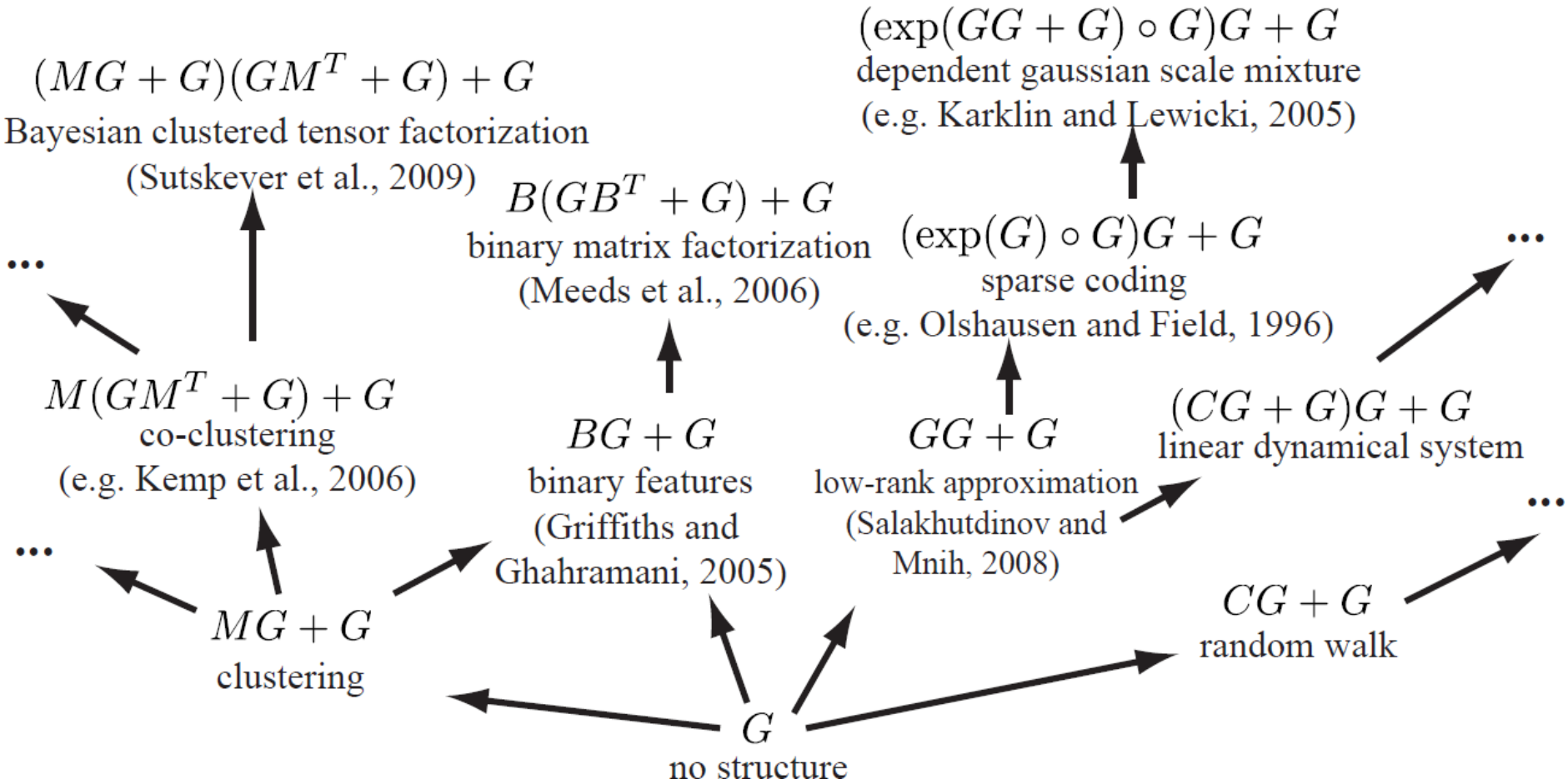
$$u_{ij} = \mathbf{1}_{i \geq j}.$$

   This is useful for modeling temporal structure, as multiplying by this matrix has the effect of cumulatively summing the rows. (Mnemonic: C for "cumulative.")

Grammar

low-rank approximation	$G \to GG + G$	(1)
clustering	$G \to MG + G \mid GM^T + G$	(2)
	$M \to MG + G$	(3)
linear dynamics	$G \to CG + G \mid GC^T + G$	(4)
sparsity	$G \to \exp(G) \circ G$	(5)
binary factors	$G \to BG + G \mid GB^T + G$	(6)
	$B \to BG + G$	(7)
	$M \to B$	(8)

Models


$(MG + G)(GM^T + G) + G$
Bayesian clustered tensor factorization
(Sutskever et al., 2009)
$(\exp(GG + G) \circ G)G + G$
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)
...
...
$M(GM^T + G) + G$
co-clustering
(e.g. Kemp et al., 2006)
$BG + G$
binary features
(Griffiths and
Ghahramani, 2005)
$GG + G$
low-rank approximation
(Salakhutdinov and
Mnih, 2008)
$(CG + G)G + G$
linear dynamical system
...
...
$MG + G$
clustering
$CG + G$
random walk
G
no structure

Inference: Individual Models

- Initialize state using one-shot algorithm for each rule application
- Latent dimensionality is determined during initialization using BNP
- Then run simple Gibbs sampler (no details provided . . .)

Initialization

1. **Low rank.** To apply the rule $G \to GG + G$, we fit the probabilistic matrix factorization (Salakhutdinov and Mnih, 2008) model using block Gibbs sampling over the two factors. While PMF assumes a fixed latent dimension, we choose the dimension automatically by placing a Poisson prior on the dimension and moving between states of differing dimension using reversible jump MCMC (Green, 1995).

2. **Clustering.** To apply the clustering rule to rows: $G \to MG + G$, or to columns: $G \to GM^T + G$, we perform collapsed Gibbs sampling over the cluster assignments in a Dirichlet process mixture model.

3. **Binary factors.** To apply the rule $G \to BG + G$ or $G \to GB^T + G$, we perform accelerated collapsed Gibbs sampling (Doshi-Velez and Ghahramani, 2009) over the binary variables in a linear-Gaussian Indian Buffet Process (Griffiths and Ghahramani, 2005) model, using split-merge proposals (Meeds et al., 2006) to escape local modes.

4. **Markov chains.** The rule $G \to CG + G$ is equivalent to estimating the state of a random walk given noisy observations, which is done using Rauch-Tung-Striebel (RTS) smoothing.

Scoring Candidate Structures

- Criterion used: predictive likelihood of held-out rows and columns
  - Marginal likelihood not feasible
  - MSE not selective enough
- Use a (stochastic) lower bound on predictive likelihood, computed using a variational approximation combined with annealed importance sampling (this is about as much detail as is in the paper . . .)

Search Over Structures

- Greedy search following grammar
  1. Start with G
  2. Expand using all possible rules
  3. Fit & score models
  4. Keep top K models
  5. Go to 2
- Assumes that good simple models will lead to good more complex models when refined
- Assumption seems to be warranted: $K = 3$ yields the same results as $K = 1$ in experiments

Results on Synthetic Data

	— 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$BG + G$

Table 1: The structures learned from 200×200 matrices generated from various distributions, with signal variance 1 and noise variance σ^2. Incorrect structures are marked with a 1, 2, or 3, depending how many decisions would need to be changed to find the correct structure. We observe that our approach typically finds the correct answer in low noise settings and backs off to simpler models in high noise settings.

Results on Real Data

	Level 1	Level 2	Level 3
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	$GM^T + G$	$(MG + G)M^T + G$	—

Results on Real Data

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
11. Tools. knife, chisel, hammer, pliers, saw, screwdriver, screw, dagger, spear, hoe, needle
12. Stuff. speck, gravel, soil, tear, bubble, slush, rust, fat, garbage, crumb, eyelash
13. Furniture. bed, chair, desk, dresser, table, sofa, seat, ladder, mattress, handrail, bench, locker
14. Liquids. wax, honey, pint, disinfectant, gas, drink, milk, water, cola, paste, lemonade, lotion
15. Structural features. bumper, cast, fence, billboard, guardrail, axle, deck, dumpster, windshield
16. Non-solid things. surf, fire, lightning, sky, steam, cloud, dance, wind, breeze, tornado, sunshine
17. Transportation. airplane, car, train, truck, jet, sedan, submarine, jeep, boat, tractor, rocket
18. Herbivores. cow, horse, lamb, camel, pig, hog, calf, elephant, cattle, giraffe, yak, goat
19. Internal organs. rib, lung, vein, stomach, heart, brain, smile, blood, lap, nerve, lips, wink
20. Carnivores. bear, walrus, shark, crocodile, dolphin, hippo, gorilla, hyena, rhinocerous

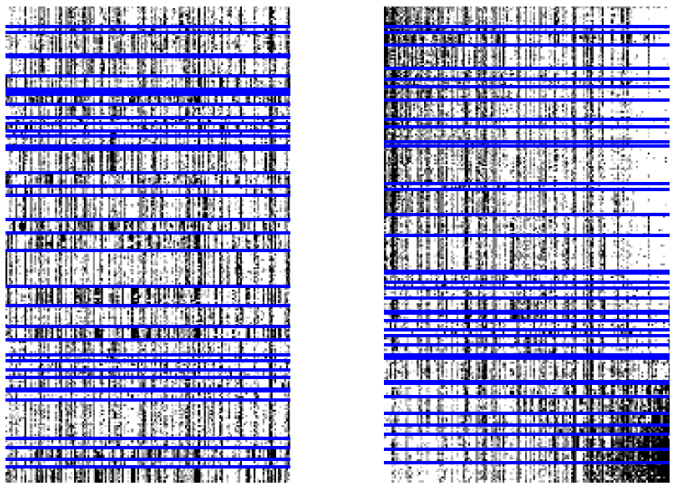
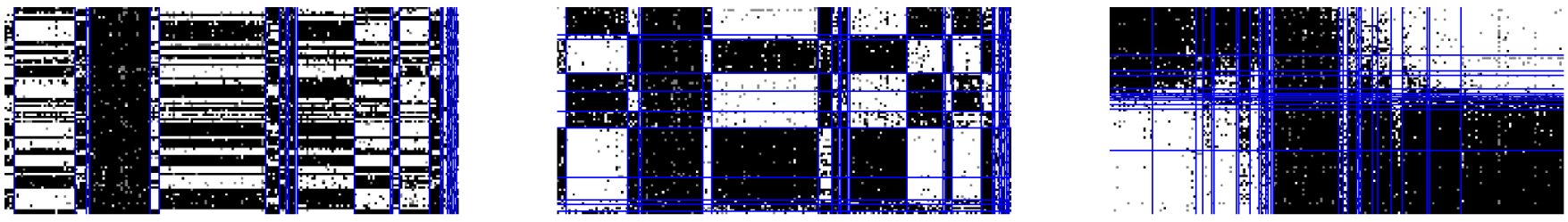


Figure 3: **(left)** The 20 largest clusters discovered by our Level 2 model $M(GG + G) + G$ for the 20 Questions dataset. Each line gives **our interpretation**, followed by random items from the cluster. **(right)** Visualizations of the Level 1 representation $MG + G$ and the Level 2 representation $M(GG + G) + G$. Rows = entities, columns = questions. 250 rows and 150 columns were selected at random from the original matrix. Rows and columns are sorted first by cluster, then by the highest variance dimension of the low-rank representation (if applicable). Clusters were sorted by the same dimension as well. Blue = cluster boundaries.

Results on Real Data



(a) Level 1: $GM^T + G$ (b) Level 2: $(MG + G)M^T + G$ (c) Level 3: $(MG + G)(GM^T + G) + G$

Figure 4: Visualization of the representations learned from the Senate voting data. Rows = Senators, columns = votes. 200 columns were selected at random from the original matrix. Black = yes, white = no, gray = absence. Blue = cluster boundaries. Rows and columns are sorted first by cluster (if applicable), then by the highest variance dimension of the low-rank representation (if applicable). Clusters are sorted by the same dimension as well. The models in the sequence increasingly reflect the polarization of the Senate.

Computing Predictive Likelihood

First, by expanding out the products in the expression, we can write the decomposition uniquely in the form

$$X = U_1 V_1 + \cdots + U_n V_n + E, \tag{1}$$

where E is an *i.i.d.* Gaussian "noise" matrix and the U_i's are any of the following: (1) a component matrix G, M, or B, (2) some number of Cs followed by G, (3) a Gaussian scale mixture. The held-out row x can therefore be represented as:

$$x = V_1^T u_1 + \cdots + V_n^T u_n + e. \tag{2}$$

The predictive likelihood is given by:

$$p(x|X_O) = \int p(U_O, V|X_O) p(u|U_O) p(x|u, V)\, dU_O\, du\, dV \tag{3}$$

where U_O is shorthand for $(U_{O1}, \ldots, U_{On})$ and u is shorthand for $(u_1, \ldots, u_n)$.

In order to evaluate this integral, we generate samples from the posterior $p(U_O, V|X)$ using the techniques described in Section 4, and compute the sample average of

$$p_{pred}(x) \triangleq \int p(u|U_O) p(x|u, V)\, du \tag{4}$$

If the term U_i is a Markov chain, the predictive distribution $p(u_i|U_O)$ can be computed using Rauch-Tung-Striebel smoothing; in the other cases, u and U_O are related only