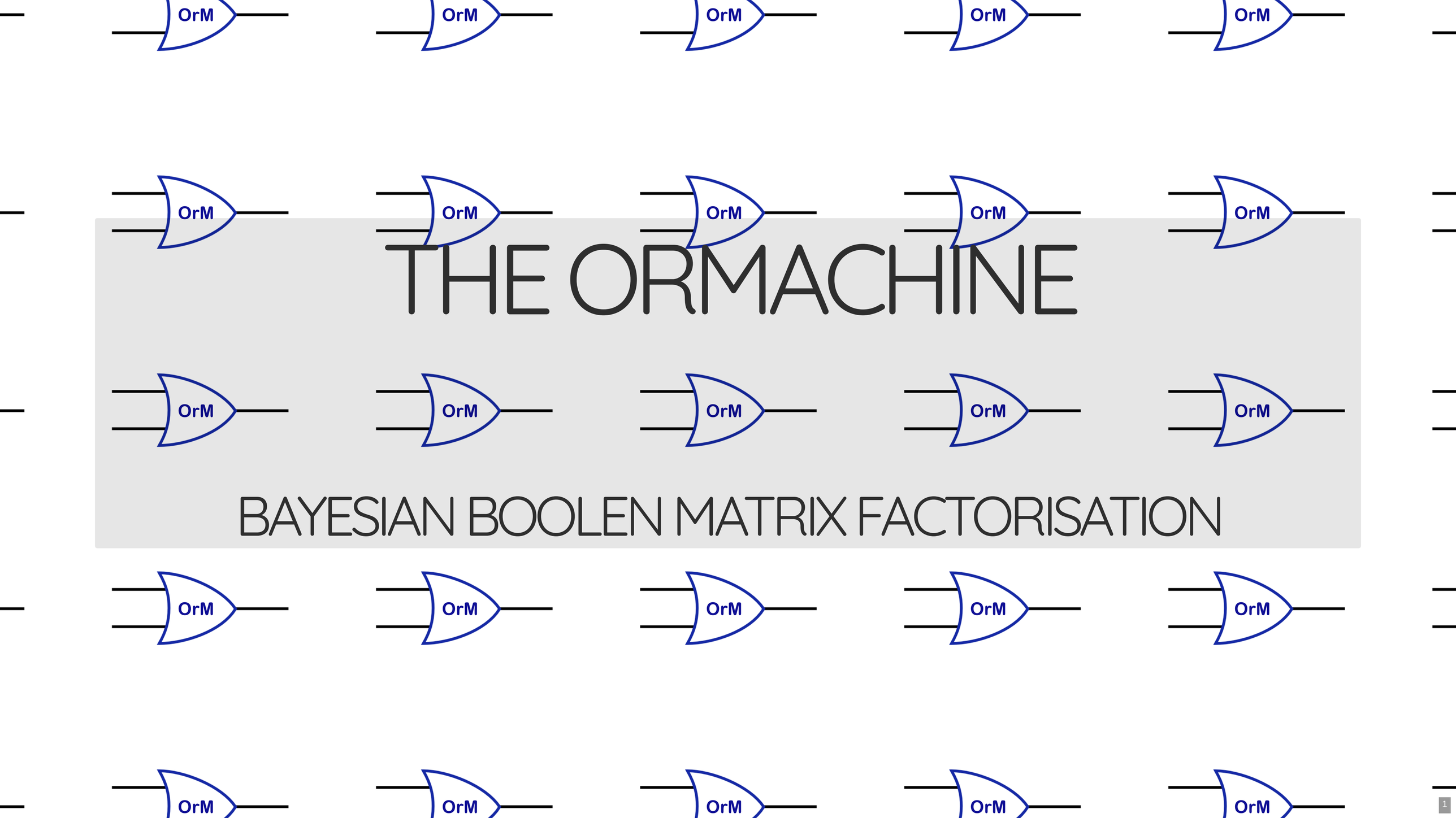
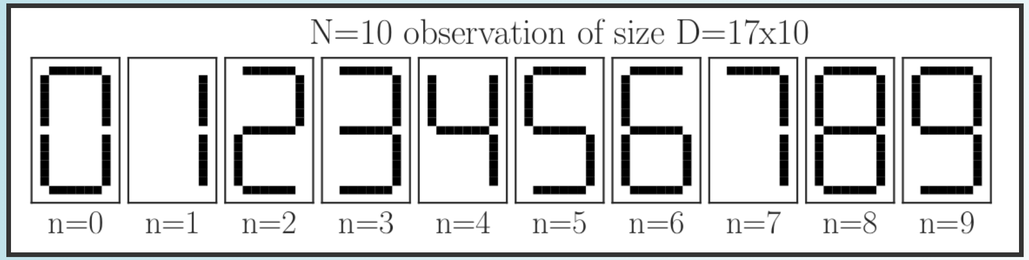


THE ORMACHINE

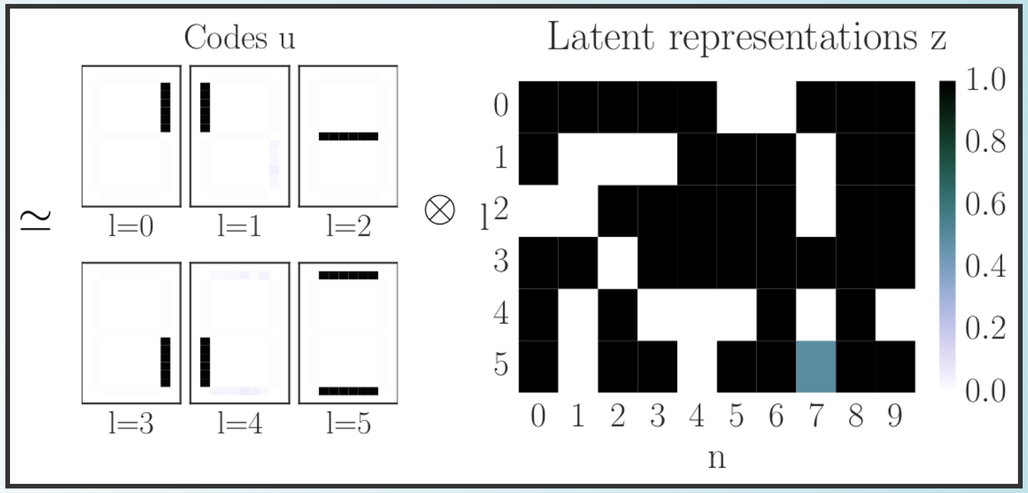
BAYESIAN BOOLEN MATRIX FACTORISATION

BOOLEAN MATRIX FACTORISATION

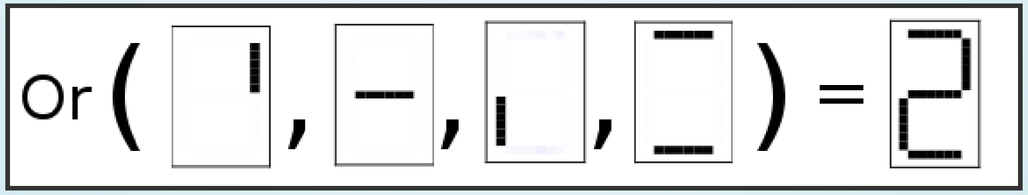
Observed Data



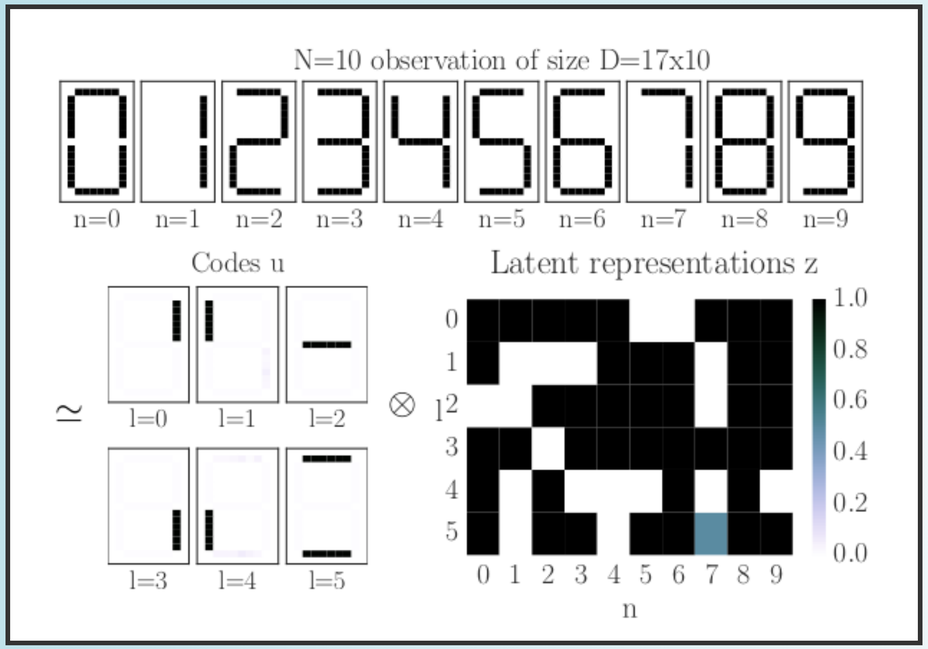
Factorisation



Example



PROBABILISTIC GENERATIVE MODEL



Notation

- x_{nd} – observations
- u_{ld} – codes (globale variables)
- z_{nl} – latent variables (local variables)
- $\lambda \geq 0$ – global noise parameter

Definitions

- Mapping $\{0, 1\}$ to $\{-1, 1\}$: $\tilde{x} = 2x - 1$
- Logistic sigmoid: $\sigma(x) = (1 + \exp[-x])^{-1}$

$$p(x_{nd}|\mathbf{u}_d, \mathbf{z}_n, \lambda) = \begin{cases} \sigma[\lambda]; & \text{if } x_{nd} = \min(1, \mathbf{z}_n^T \mathbf{u}_d) \\ 1 - \sigma[\lambda] = \sigma[-\lambda]; & \text{if } x_{nd} \neq \min(1, \mathbf{z}_n^T \mathbf{u}_d) \end{cases}$$

$$= \sigma\left[\lambda \tilde{x}_{nd} \left(1 - 2 \prod_l (1 - z_{nl} u_{ld})\right)\right]$$

INFERENCE FOR THE ORMACHINE

FULL CONDITIONALS

- Likelihood:

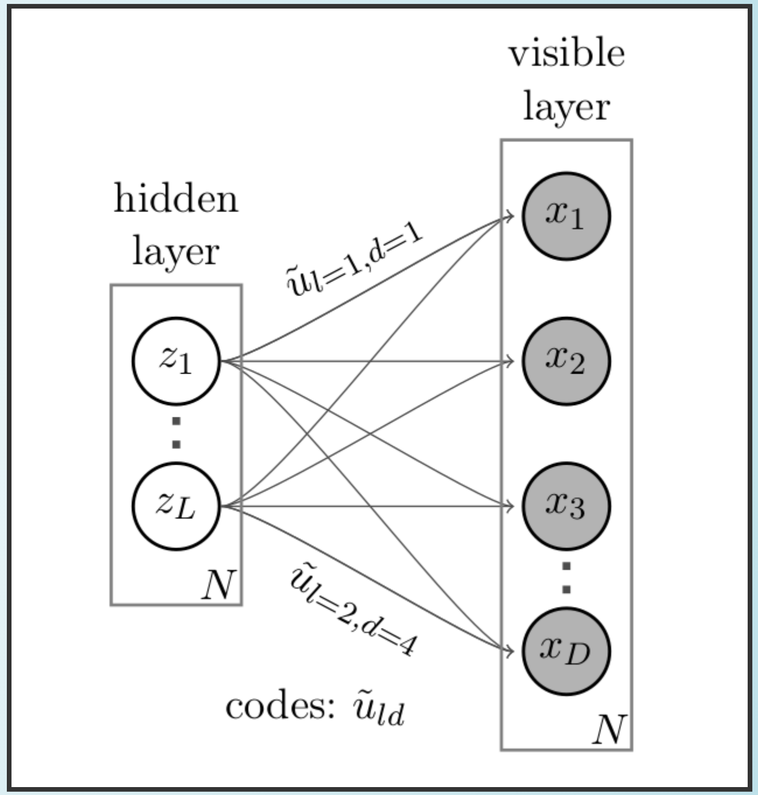
$$L = \prod_{nd} \sigma\left[\lambda \tilde{x}_{nd}(1 - 2\prod_{l}(1 - z_{nl}u_{ld})\right]$$

- Full Conditional:

$$p(z_{nl}|\text{rest}) = \sigma\left[\lambda \tilde{z}_{nl} \sum_{d} \tilde{x}_{nd}\ u_{ld} \prod_{l' \neq l}(1 - z_{nl'}u_{l'd})\right]$$

  - Intuition: Need to consider the full Markov Blanket.

- **Computational shortcut:**
  - $u_{ld} = 0 \rightarrow$ No effect of z_{nl} on the likelihood.
  - $z_{nl'}u_{l'd} = 1$ for $l' \neq l \rightarrow x_{nd}$ is **explained away**.



A MODIFIED BINARY STATE GIBBS SAMPLER

- Old state: $\mathbf{x}$, new state: $\mathbf{y}$.
- Gibbs sampler: Draw a new value from the full conditional $p(y|\text{rest})$.
- Here, we propose value y **different from the** x with probability 1.
- Metropolis-Hasting:

$$p(\text{accept}) = p(\text{mutate}) = \frac{p(y|\text{rest})q(x|y)}{p(x|\text{rest})q(y|x)} = \frac{p(y|\text{rest})}{1 - p(y|\text{rest})} \geq p(y|\text{rest})$$

- Typical Gibbs sampler:

- Metropolised Gibbs sampler:

DISPERSION PARAMTER λ

- How many entries are correctly predicted by the deterministic Boolean product

$$P = \sum_{n,d} I \left[x_{nd} = (1 - 2 \prod_{l} (1 - z_{nl} u_{ld}) \right]$$

- We can rewrite the likelihood

$$L = \sigma(\lambda)^P \sigma(-\lambda)^{(ND-P)}$$

- We find the MLE of $\sigma(\lambda)$ in **closed form**:

$$\sigma(\lambda)_{\text{mle}} = \frac{P}{ND} \ .$$

METROPOLISED GIBBS SAMPLER - ALGORITHM

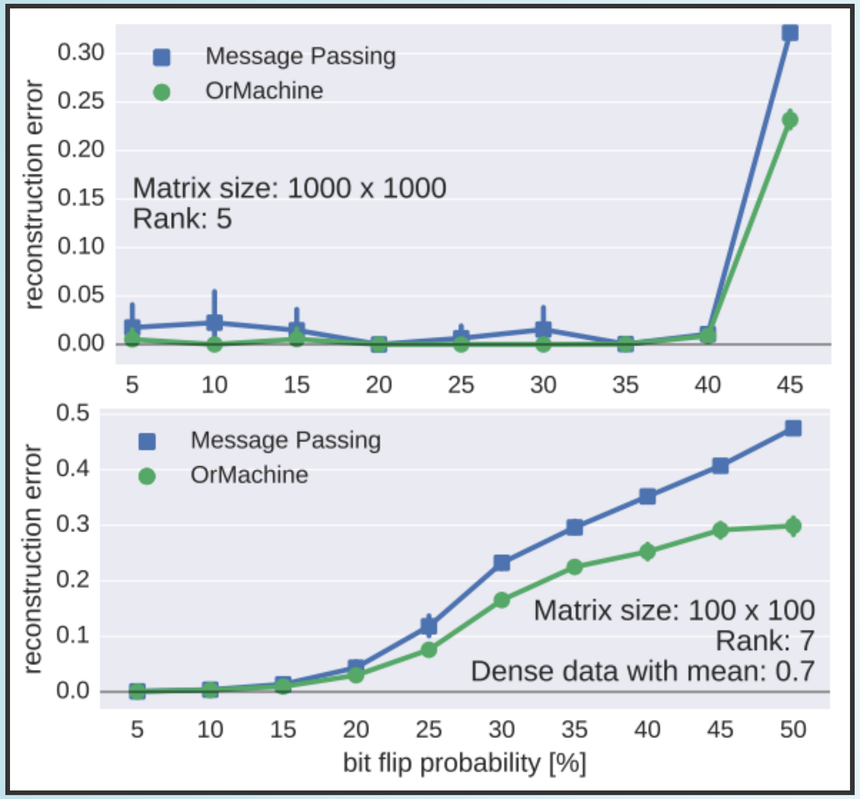
Algorithm 2 Sampling from the OrMachine

for i in $1, \ldots,$ max-iters **do**
  for n in $1, \ldots,$ N **do**
    for l in $1, \ldots,$ L **do**
      Compute $p(z_{nl}|\text{rest})$
      Flip z_{nl} with probability $[p(z_{nl}|\text{rest})^{-1} - 1]^{-1}$
    end for
  end for
  for d in $1, \ldots,$ d **do**
    for l in $1, \ldots,$ L **do**
      Compute $p(u_{ld}|\text{rest})$
      Flip u_{ld} with probability $[p(u_{ld}|\text{rest})^{-1} - 1]^{-1}$
    end for
  end for
  Set λ to its MLE
end for

EXAMPLES AND EXPERIMENTS

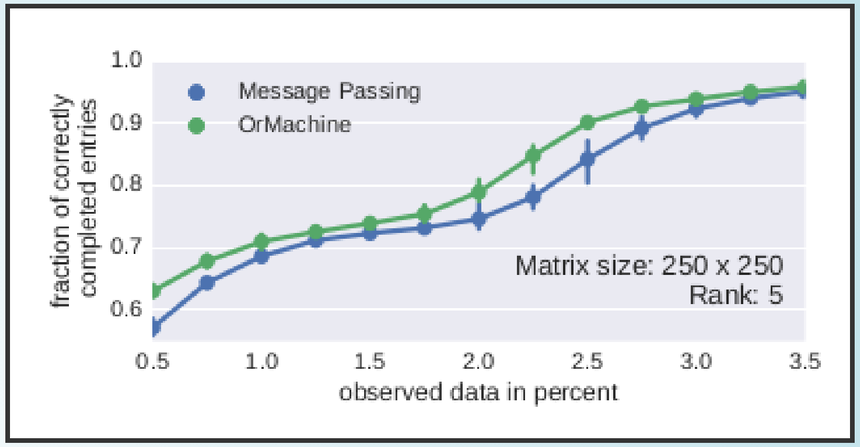
SYNTHETIC DATA BENCHMARKS

Random Matrix Factorisation

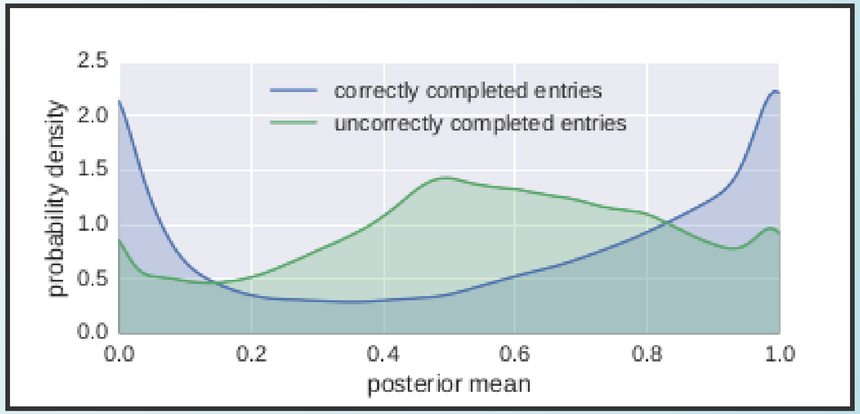


[Message Passing arXiv: 1509.08535]

Random Matrix Completion

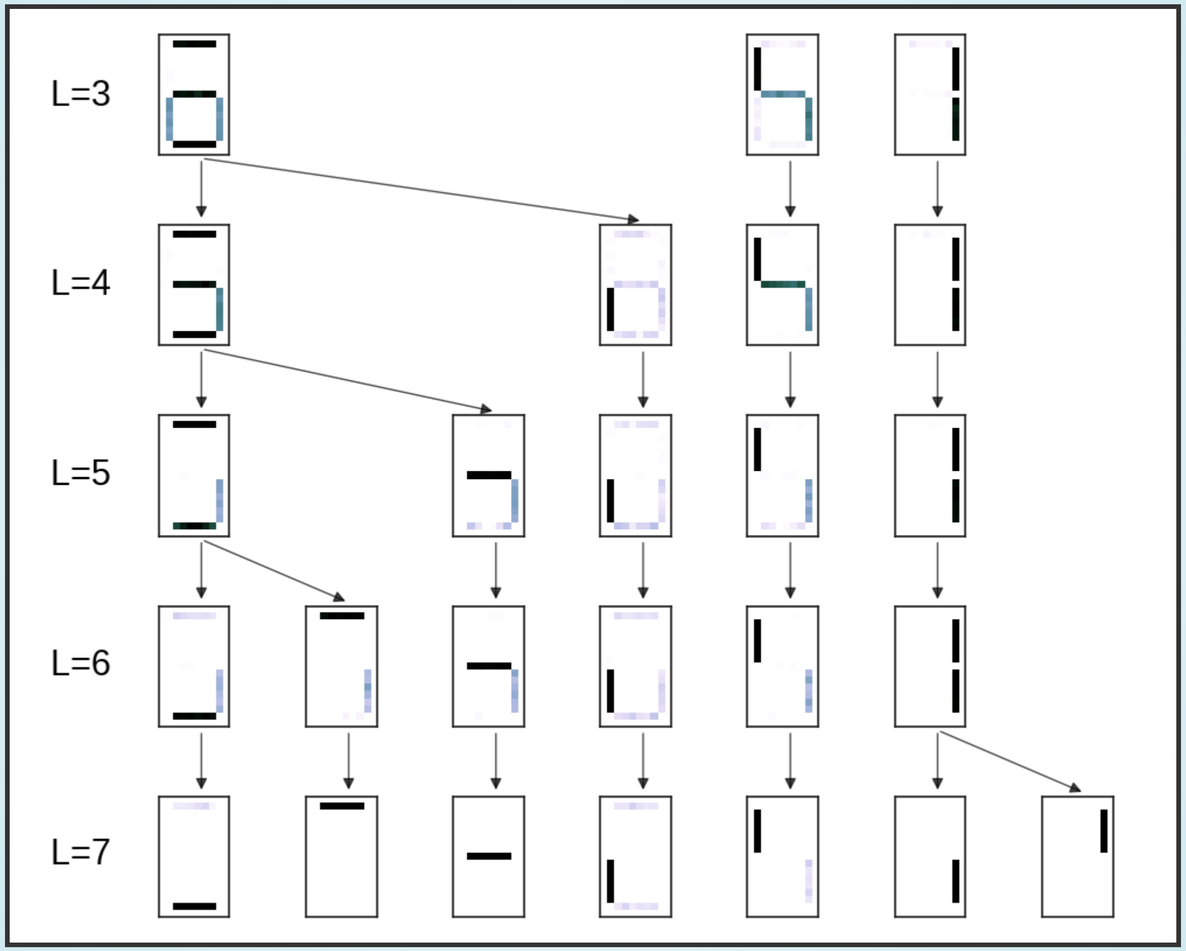


Density of posterior means



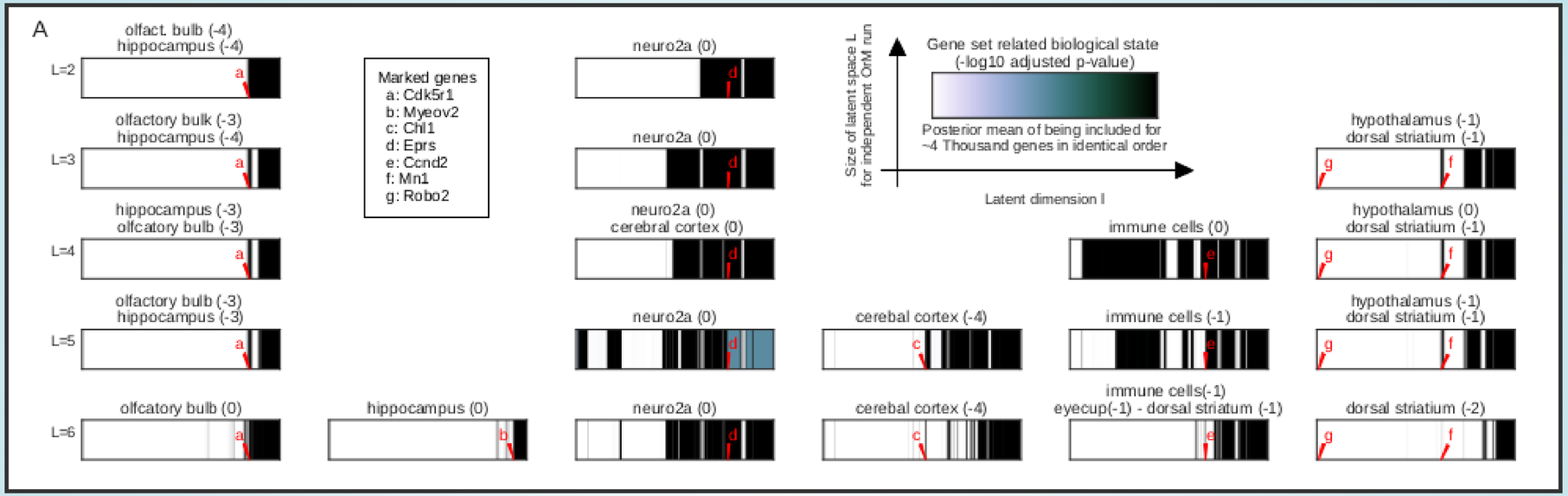
SINGLE CELL DATA II - 1.3 MILLION BRAIN CELLS X 20K GENES (E18 MICE)

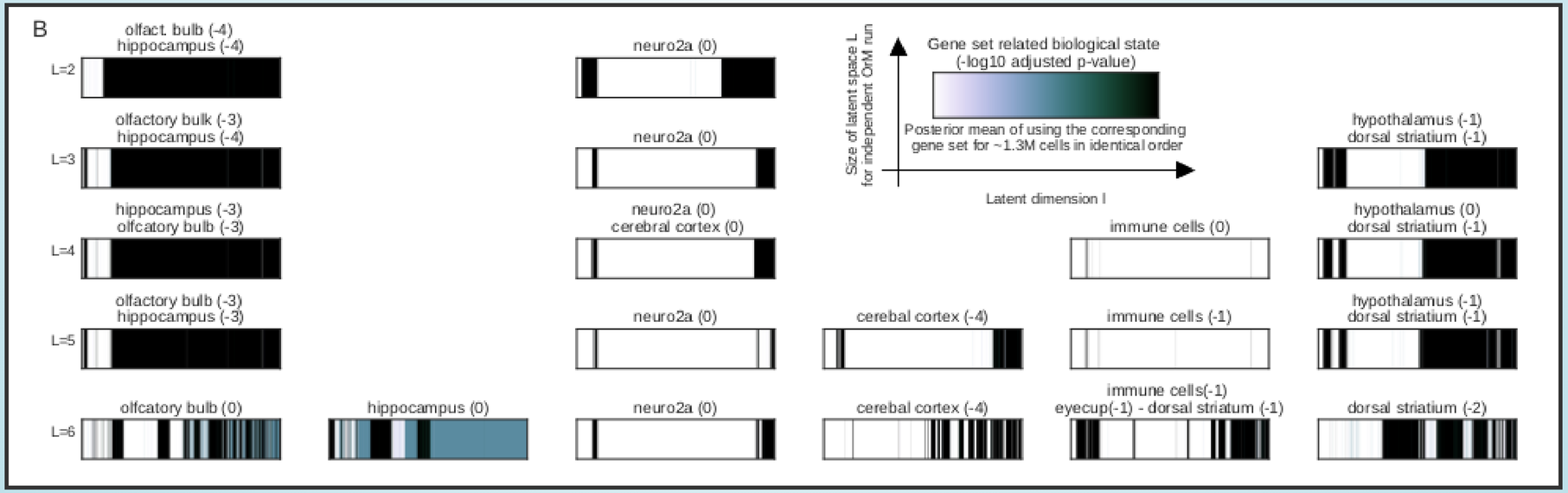
CALCULATOR DIGIT HIERARCHY



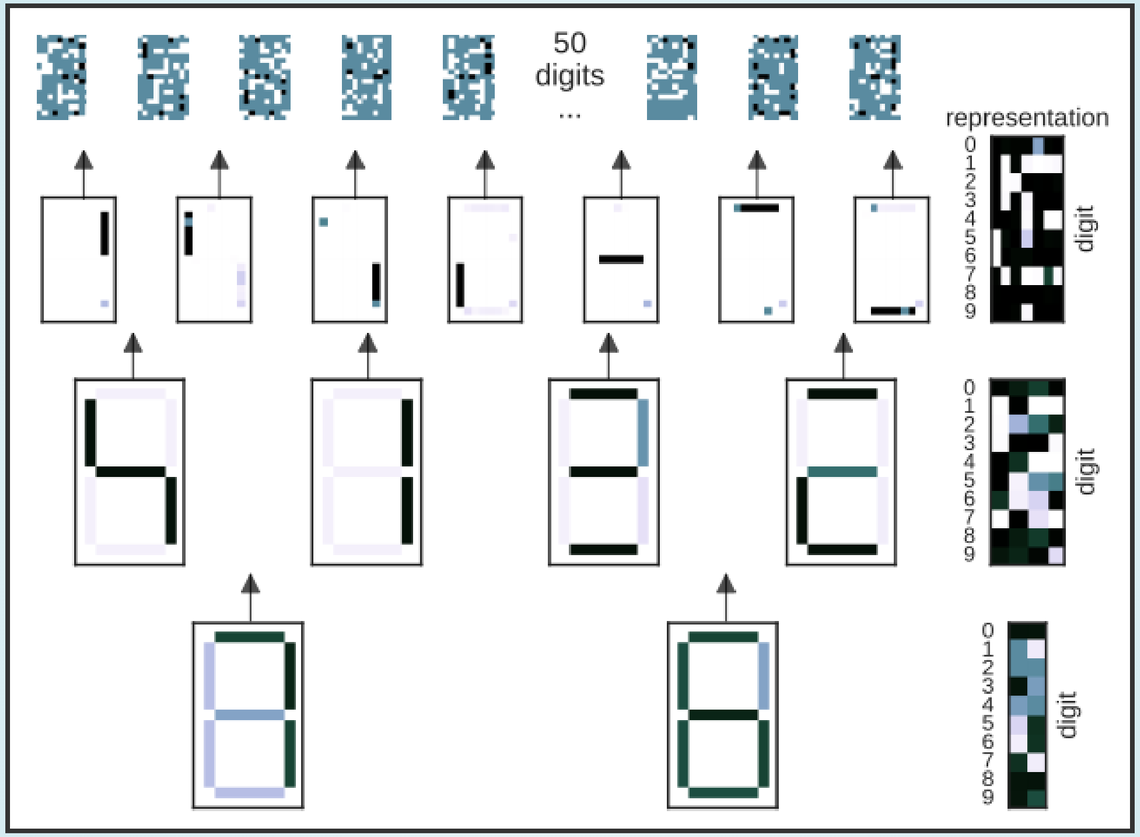
- OrMachines of different dimensionality on noise-free calculator digits

GENE PATTERNS – CELL REPRESENTATIONS





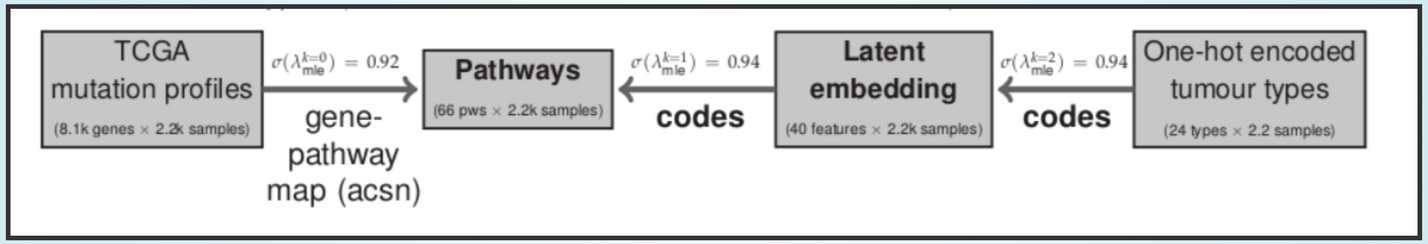
DEEP NOISY CALCULATOR DIGITS



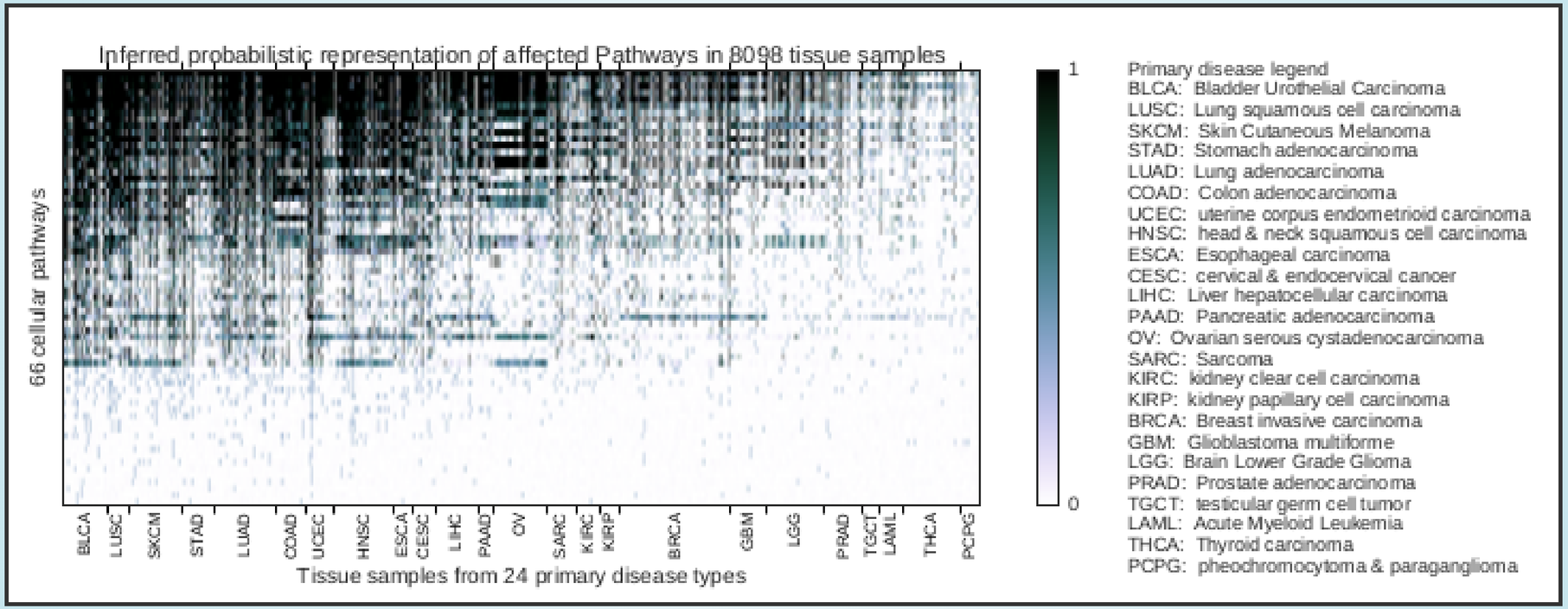
- Input: 50 digits with 70% missing observations
- Reduce reconstruction error from 1.4% to 0.4% compared to shallow model

PANCAN AND PATHWAY DATA

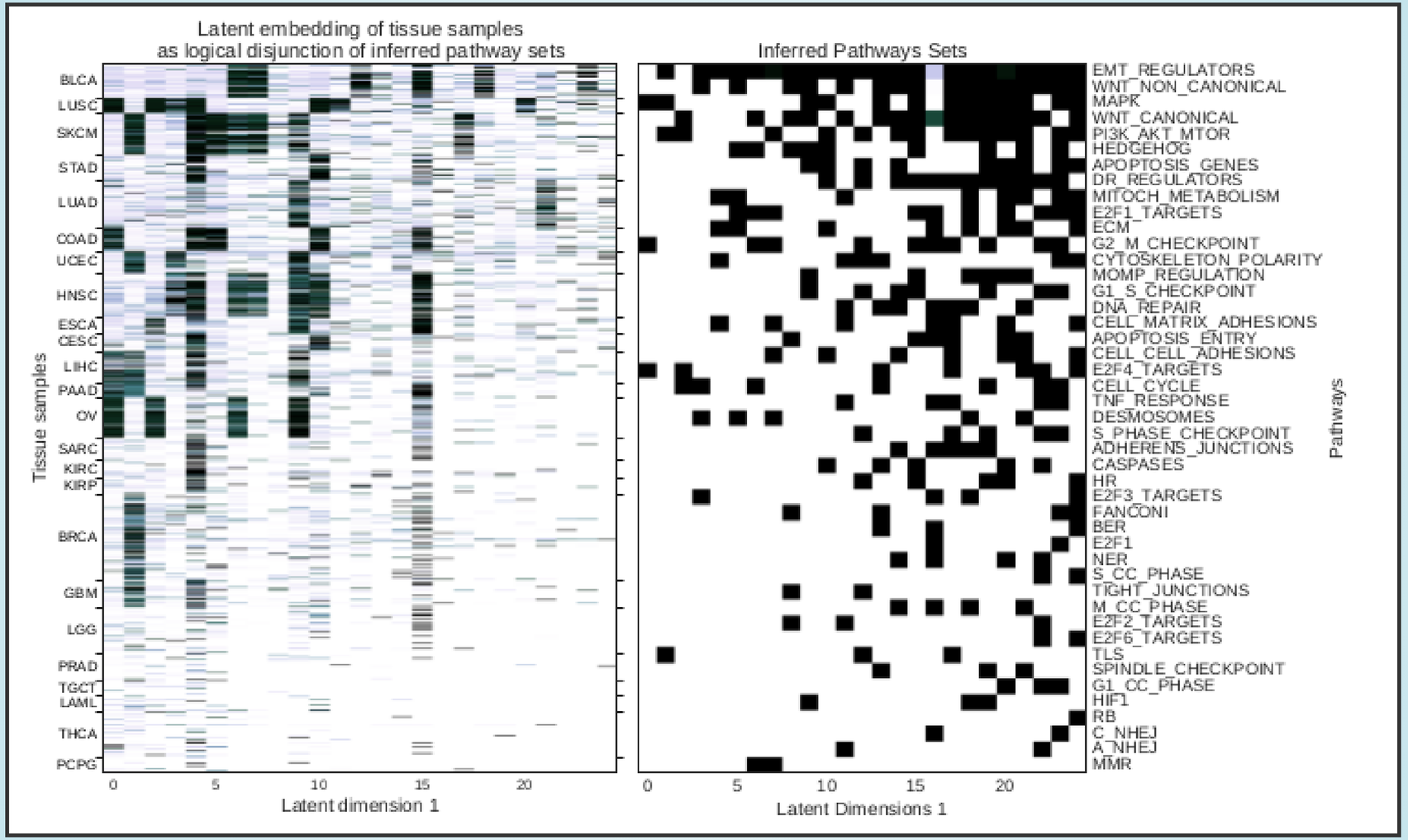
SETUP: COMBINE LAYERS OF ORMACHINES



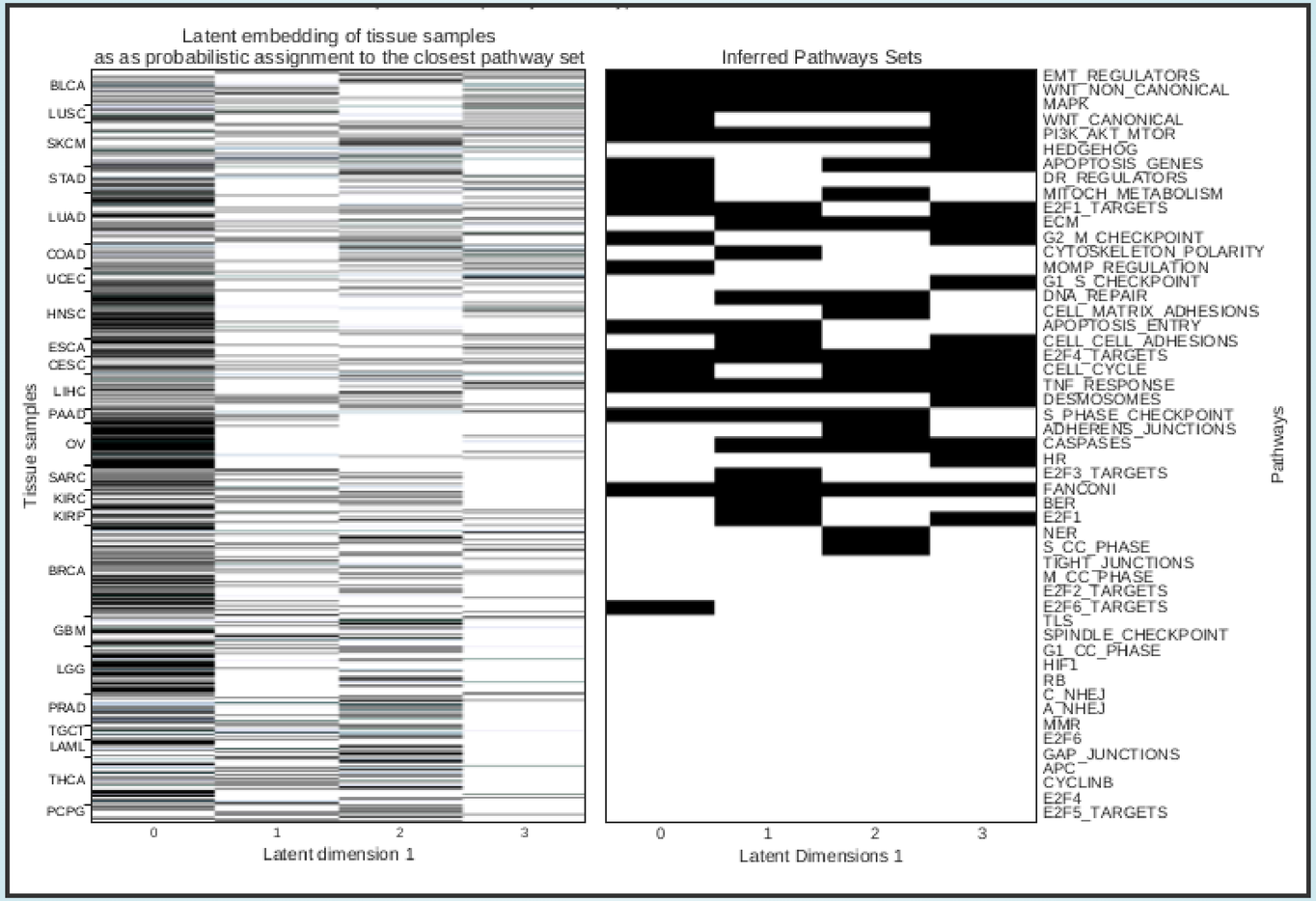
"DATA"


Inferred probabilistic representation of affected Pathways in 8098 tissue samples
66 cellular pathways
BLCA
LUSC
SKCM
STAD
LUAD
COAD
UCEC
HNSC
ESCA
CESC
LIHC
PAAD
OV
SARC
KIRC
KIRP
BRCA
GBM
LGG
PRAD
TGCT
LAML
THCA
PCPG
Tissue samples from 24 primary disease types
1
0
Primary disease legend
BLCA: Bladder Urothelial Carcinoma
LUSC: Lung squamous cell carcinoma
SKCM: Skin Cutaneous Melanoma
STAD: Stomach adenocarcinoma
LUAD: Lung adenocarcinoma
COAD: Colon adenocarcinoma
UCEC: uterine corpus endometrioid carcinoma
HNSC: head & neck squamous cell carcinoma
ESCA: Esophageal carcinoma
CESC: cervical & endocervical cancer
LIHC: Liver hepatocellular carcinoma
PAAD: Pancreatic adenocarcinoma
OV: Ovarian serous cystadenocarcinoma
SARC: Sarcoma
KIRC: kidney clear cell carcinoma
KIRP: kidney papillary cell carcinoma
BRCA: Breast invasive carcinoma
GBM: Glioblastoma multiforme
LGG: Brain Lower Grade Glioma
PRAD: Prostate adenocarcinoma
TGCT: testicular germ cell tumor
LAML: Acute Myeloid Leukemia
THCA: Thyroid carcinoma
PCPG: pheochromocytoma & paraganglioma

EMBEDDING


Latent embedding of tissue samples
as logical disjunction of inferred pathway sets
Tissue samples
BLCA
LUSC
SKCM
STAD
LUAD
COAD
UCEC
HNSC
ESCA
CESC
LIHC
PAAD
OV
SARC
KIRC
KIRP
BRCA
GBM
LGG
PRAD
TGCT
LAML
THCA
PCPG
0
5
10
15
20
Latent dimension 1
Inferred Pathways Sets
EMT_REGULATORS
WNT_NON_CANONICAL
MAPK
WNT_CANONICAL
PI3K_AKT_MTOR
HEDGEHOG
APOPTOSIS_GENES
DR_REGULATORS
MITOCH_METABOLISM
E2F1_TARGETS
ECM
G2_M_CHECKPOINT
CYTOSKELETON_POLARITY
MOMP_REGULATION
G1_S_CHECKPOINT
DNA_REPAIR
CELL_MATRIX_ADHESIONS
APOPTOSIS_ENTRY
CELL_CELL_ADHESIONS
E2F4_TARGETS
CELL_CYCLE
TNF_RESPONSE
DESMOSOMES
S_PHASE_CHECKPOINT
ADHERENS_JUNCTIONS
CASPASES
HR
E2F3_TARGETS
FANCONI
BER
E2F1
NER
S_CC_PHASE
TIGHT_JUNCTIONS
M_CC_PHASE
E2F2_TARGETS
E2F6_TARGETS
TLS
SPINDLE_CHECKPOINT
G1_CC_PHASE
HIF1
RB
C_NHEJ
A_NHEJ
MMR
Pathways
0
5
10
15
20
Latent Dimensions 1

CLUSTERING VIA ONE-HOT ACTIVATIONS

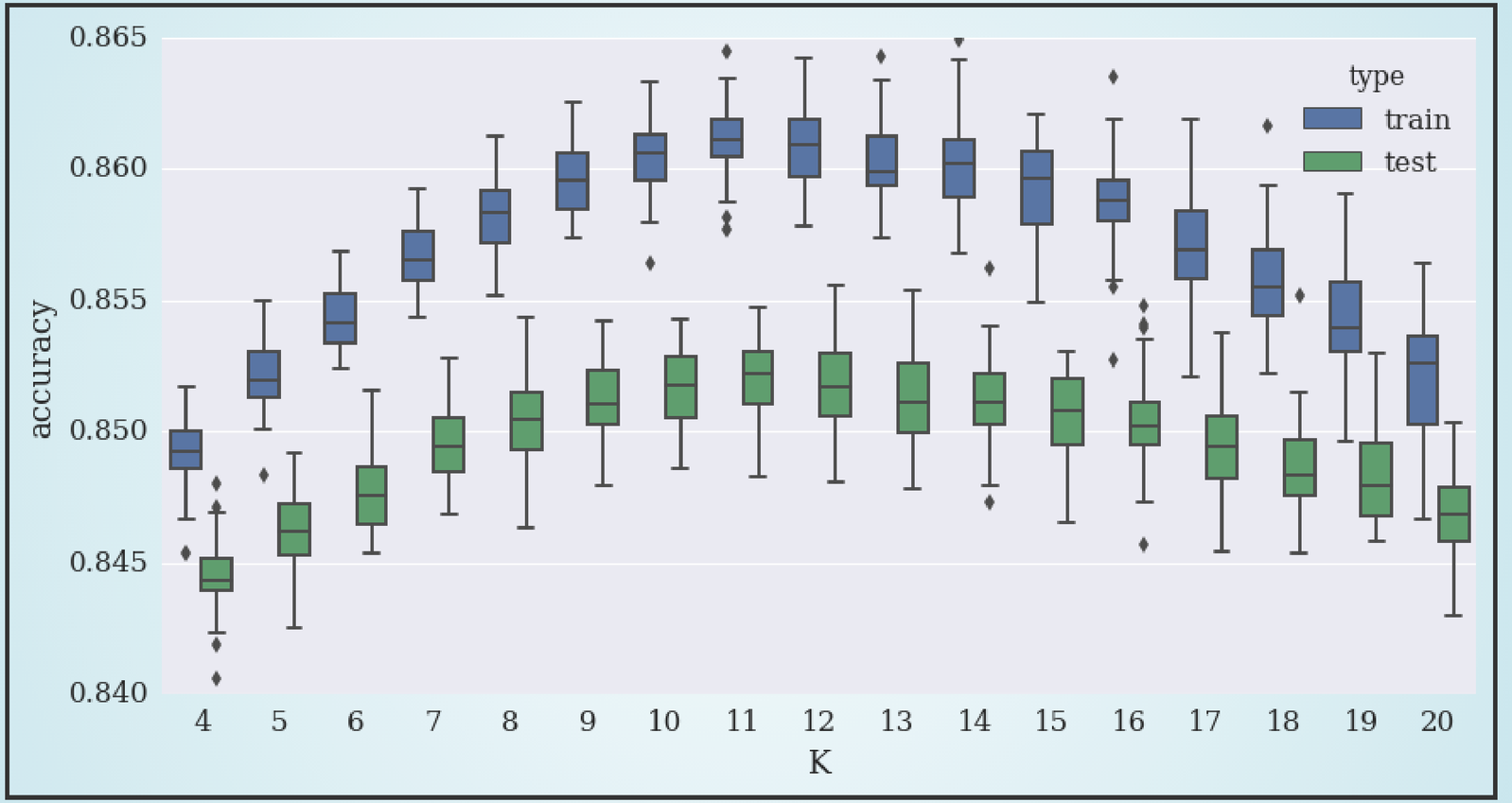


CONCLUSION

- Boolean Matrix Factorisation is a **simple and intuitive** model for binary observations.
- The OrMachine: a **probabilistic** model with highly **scalable Bayesian inference**.
- Outperforms state of the methods for Boolean matrix factorisation and completion.
- Missing data and prior knowledge can easily be dealt with.
- Multiple-layers of factorisation can extract remaining structure, build interpretable hierarchies of abstraction and leverage on additional prior information

ADDITIONAL MATERIAL

AUTO-REGULATING SPARSITY



PREPRINT ON ARXIV

Cornell University Library

arXiv.org > stat > arXiv:1702.06166

Statistics > Machine Learning

Bayesian Boolean Matrix Factorisation

Tammo Rukat, Chris C. Holmes, Michalis K. Titsias, Christopher Yau

(Submitted on 20 Feb 2017 (v1), last revised 25 Feb 2017 (this version, v2))

Boolean matrix factorisation aims to decompose a binary data matrix into an approximate Boolean product of two low rank, binary matrices: one containing meaningful patterns, the other q of these patterns. We introduce the OrMachine, a probabilistic generative model for Boolean matrix factorisation and derive a Metropolised Gibbs sampler that facilitates efficient parallel p outperforms all currently existing approaches for Boolean matrix factorisation and completion. This is the first method to provide full posterior inference for Boolean Matrix factorisation whic in collaborative filtering and, crucially, improves the interpretability of the inferred patterns. The proposed algorithm scales to large datasets as we demonstrate by analysing single cell ger thousand genes on commodity hardware.

HAMMING MACHINE

- Construct a probability distribution based on the hamming distance between two binary vectors, $h(\mathbf{x}, \mathbf{u})$, and a dispersion parameter λ:

$$p(\mathbf{x}|\mathbf{u}) \propto \exp[-\lambda\, h(\mathbf{x}, \mathbf{u})]$$

- Each observations $\mathbf{x}$ is generated from a subset of binary **codes**: $\mathbf{u}_{l=1\ldots L}$, selected by a vector of binary latent variables $\mathbf{z}$

$$p(\mathbf{x}|\mathbf{U}, \mathbf{z}, \lambda) \propto \prod_l p(\mathbf{x}|\mathbf{u}_l, \lambda)^{z_l} = \prod_d \exp\left[-\sum_l z_l \lambda h(x_d, u_{ld})\right]$$

- Normalising the likelihood for for binary observations yields a **logistic sigmoid**:

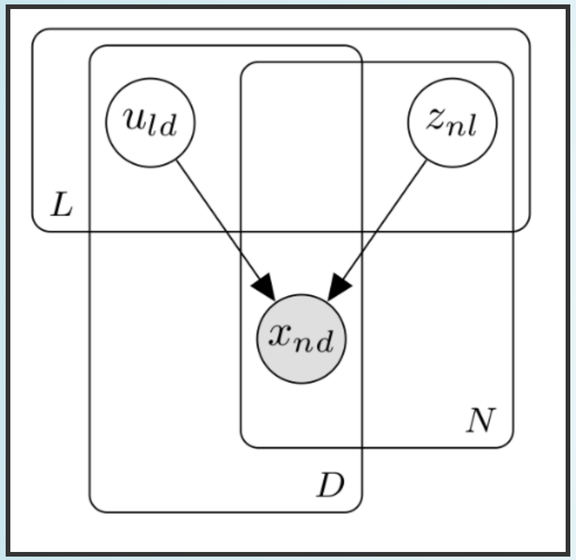
$$p(x_d = 1|\mathbf{z}, \mathbf{u}_{1\ldots L}, \lambda) = \frac{1}{1 + \exp\left[-\lambda \sum_l z_l (2u_{ld} - 1)\right]} = \sigma\left[\lambda \sum_l z_l \tilde{u}_{ld}\right]$$

- We defined the mapping from $\{0, 1\}$ to $\{-1, 1\}$: $\tilde{u} = 2u - 1$

ONE-HOT SAMPLING

INTRODUCTION TO LATENT VARIABLE MODELS

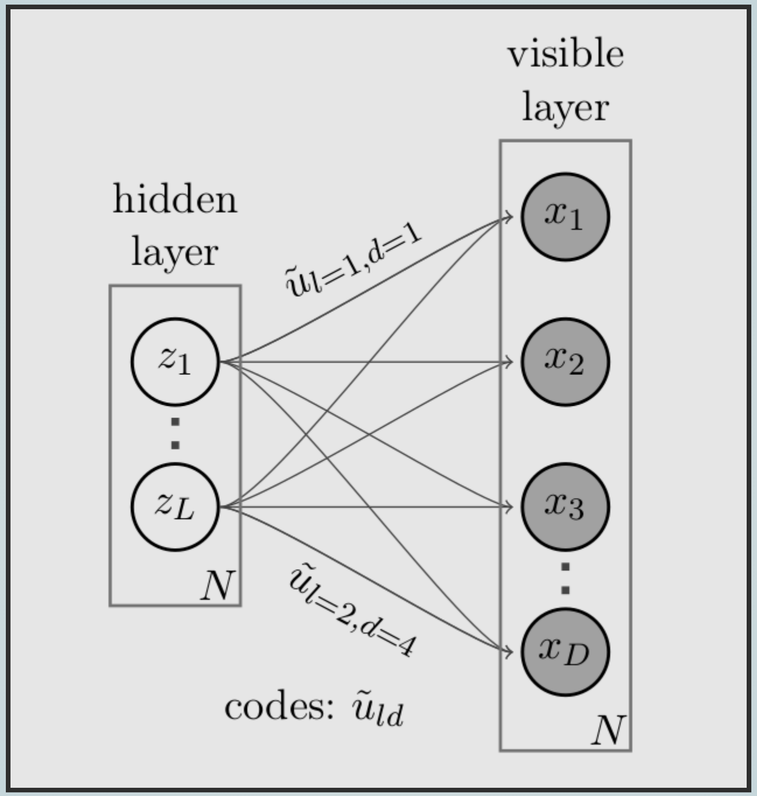
NOTATION AND GRAPHICAL MODEL



- Mixture models
- Factor Analysis (PCA)

- Variables
  - x_{nd} – observations
  - u_{ld} – parameters (globale variables, weights)
  - z_{nl} – latent variables (local variables)
- Indices
  - $n = 1 \dots N$ – observations/specimens
  - $d = 1 \dots D$ – features (e.g. pixels or genes)
  - $l = 1 \dots L$ – latent dimensions
  - $k = 1 \dots K$ – layers
- N observations
- D features
- L latent variables
- K layers / abstraction levels

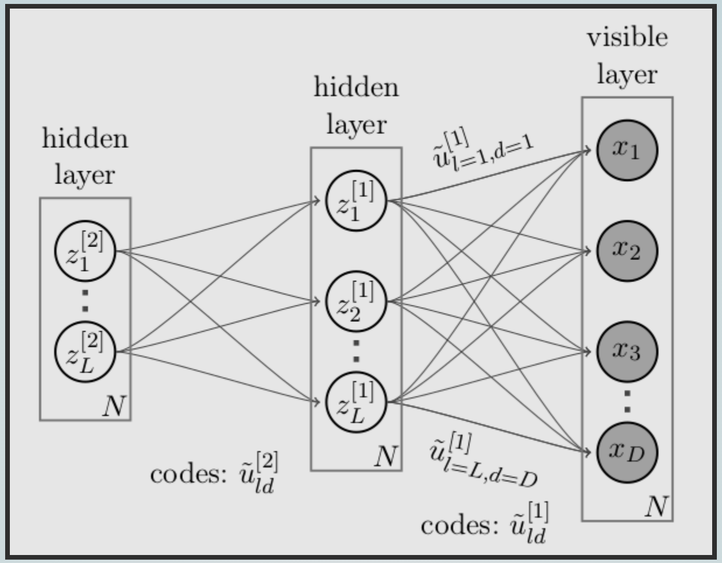
NEURAL NETWORK



- Major difference to feed forward neural nets: Nodes **and** weights are stochastic

WHAT MAKES A GOOD LATENT VARIABLE MODEL FOR BIOLOGICAL DATA?

MULTI-LAYER ORMACHINE



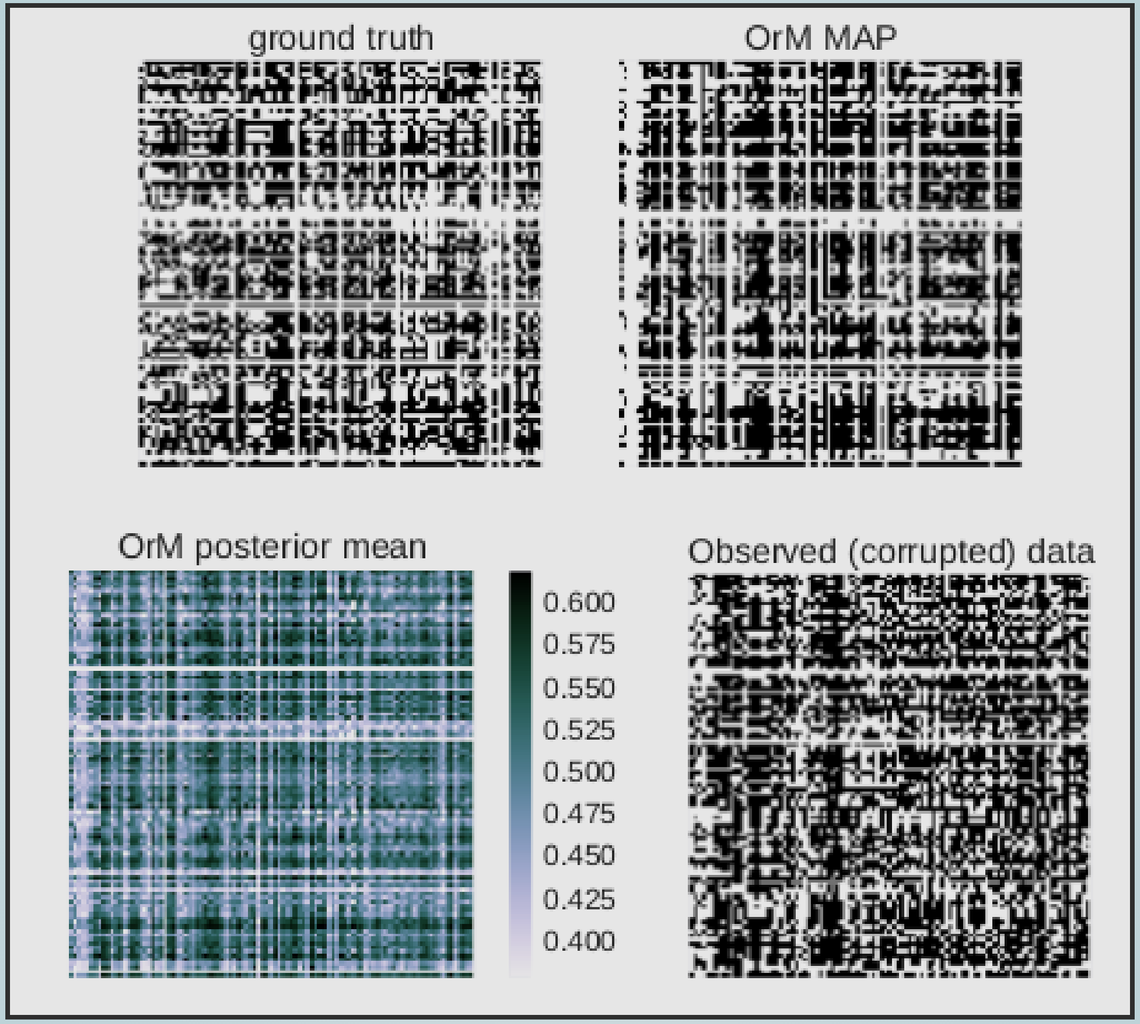
With $\mathbf{z}_n^{[0]} = \mathbf{x}_n$ and $L^{[0]} = D$, that is

$$p(\mathbf{Z}^{[0:K]}, \mathbf{U}^{[1:K]}, \lambda) = p(\mathbf{Z}^{[K]}) \prod_{k=0}^{K-1} p(\mathbf{Z}^{[k]} | \mathbf{Z}^{[k+1]}, \mathbf{U}^{[k+1]}, \lambda^{[k+1]})\, p(\mathbf{U}^{[k+1]})\, p(\lambda^{[k+1]})$$

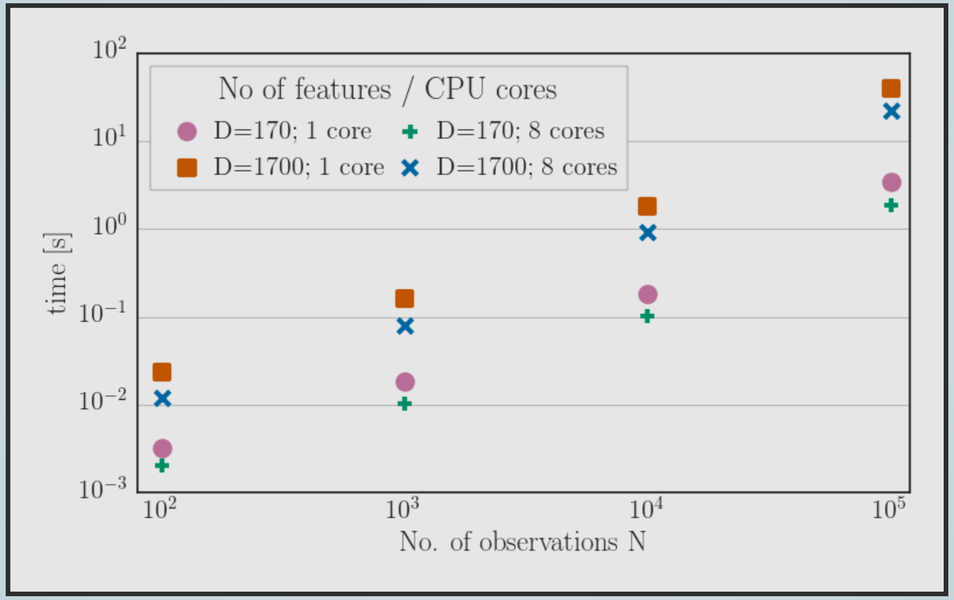
The joint density factorises in terms of the form p(layer|parents)

RANDOM MATRIX FACTORISATION

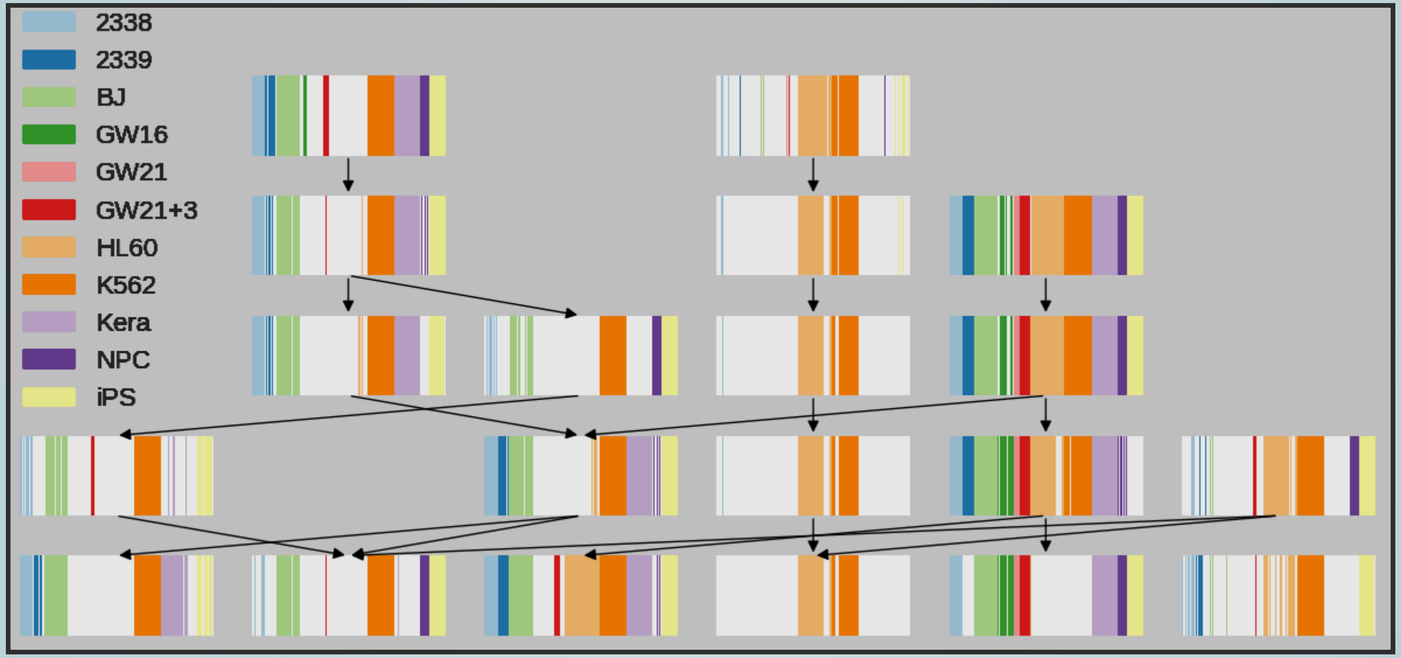
PROBLEM SETTING



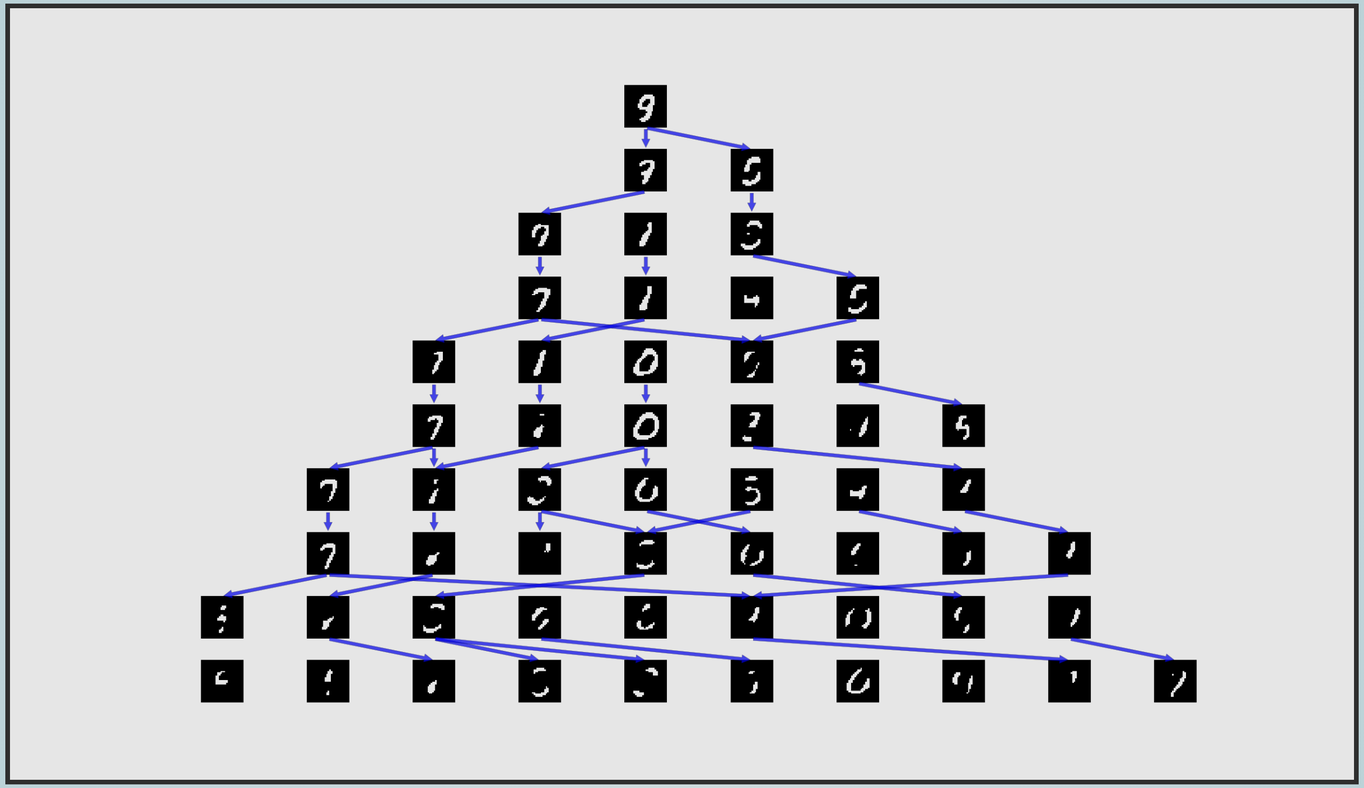
SPEED


No of features / CPU cores
D=170; 1 core
D=170; 8 cores
D=1700; 1 core
D=1700; 8 cores
time [s]
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{2}
10^{3}
10^{4}
10^{5}
No. of observations N

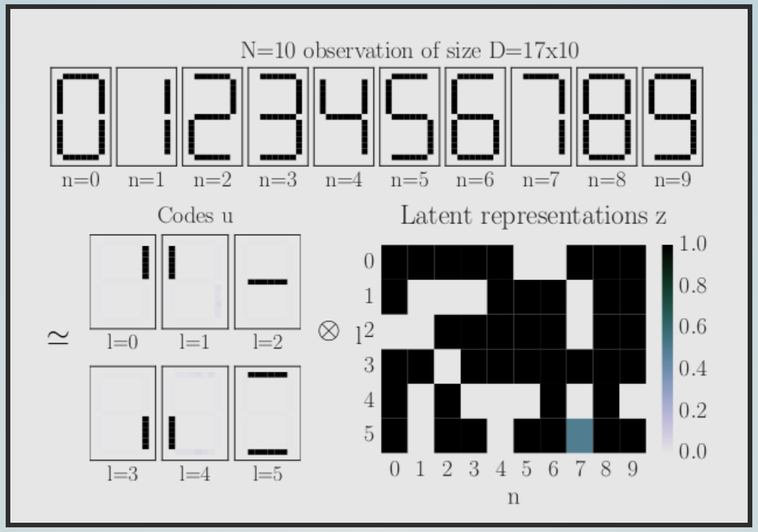
SINGLE CELL DATA I

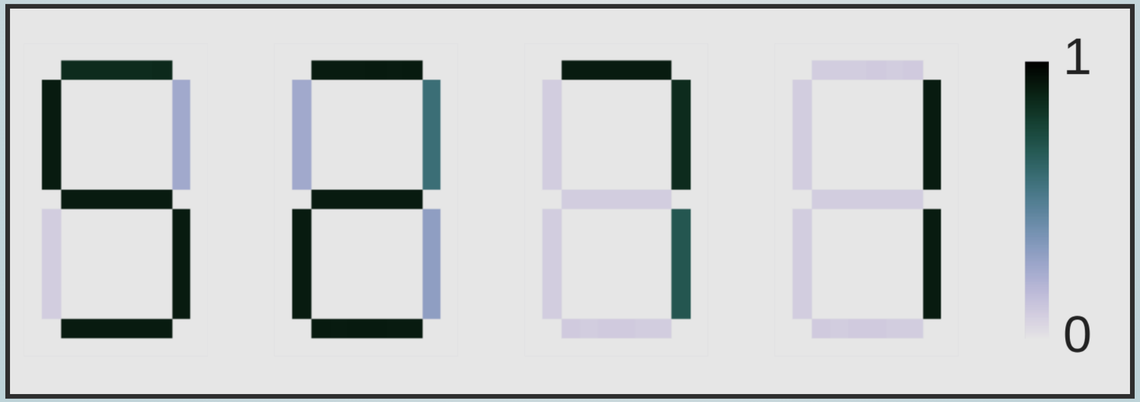


MNIST



DEEP CALCULATOR DIGITS





- Second layer representation fed forward to data layer.

A LITTLE DETOUR: PESKUN'S THEOREM

- We have
  - A random variable X following a distribution π
  - Transition matrices P_1 and P_2 that are reversible for π:
    $$\pi(x)P(x,y) = \pi(y)P(y,x)$$
  - Define $P_2 \geq P_1$, if it's true for every off-diagonal element.
- The theorem states, if
  $$P_2 \geq P_1$$
  then:
  $$v(f,\pi,P_1) \geq v(f,\pi,P_2)$$
  where
  $$v(f,\pi,P) = \lim_{N\to\infty} N\mathrm{var}(\hat{I}_N)$$
  is the variance of some estimator
  $$\hat{I}_N = \sum^{N} \frac{f(X^{(t)})}{\quad} \text{ of } I = E_\pi(f)$$

IMPLEMENTATION

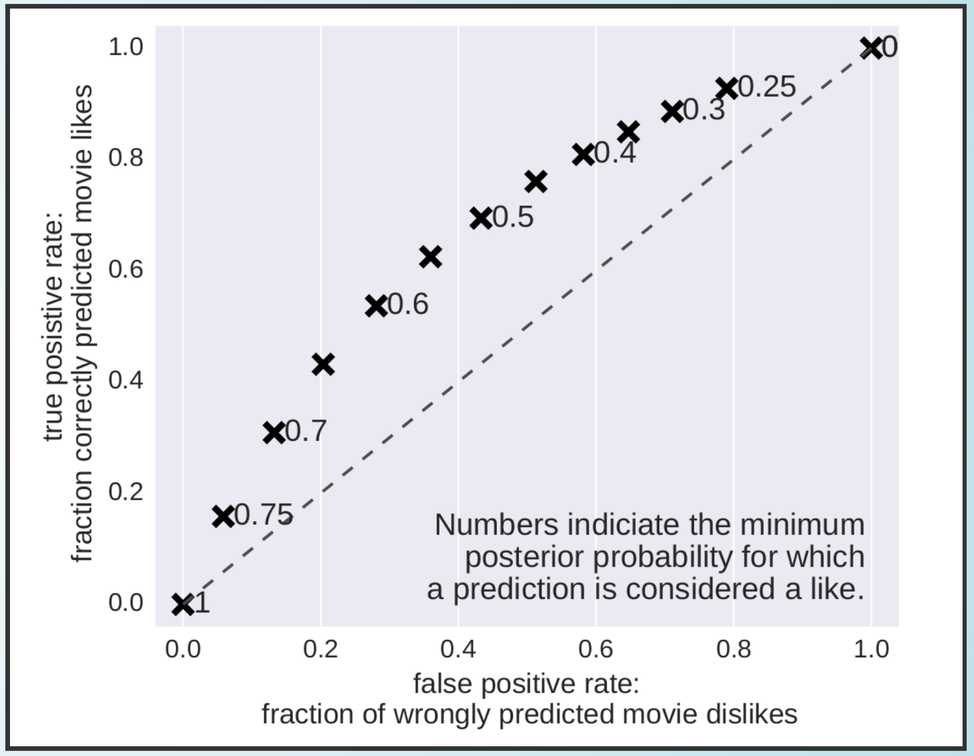
Algorithm 1 Computation of the full conditional of z_{nl}

accumulator $= 0$
for d in $1, \ldots, D$ **do**
  if $u_{ld} = 0$ **then**
    continue (next iteration over d)
  end if
  for l' in $1, \ldots, L$ **do**
    if $l' \neq l$ and $z_{nl'} = 1$ and $u_{l'd} = 1$ **then**
      continue (next iteration over d)
    end if
  end for
  accumulator $=$ accumulator $+ \tilde{x}_{nd}$
end for
$p(z_{nl} | \cdot) = \sigma\left(\lambda \cdot \tilde{z}_{nl} \cdot \text{accumulator}\right)$

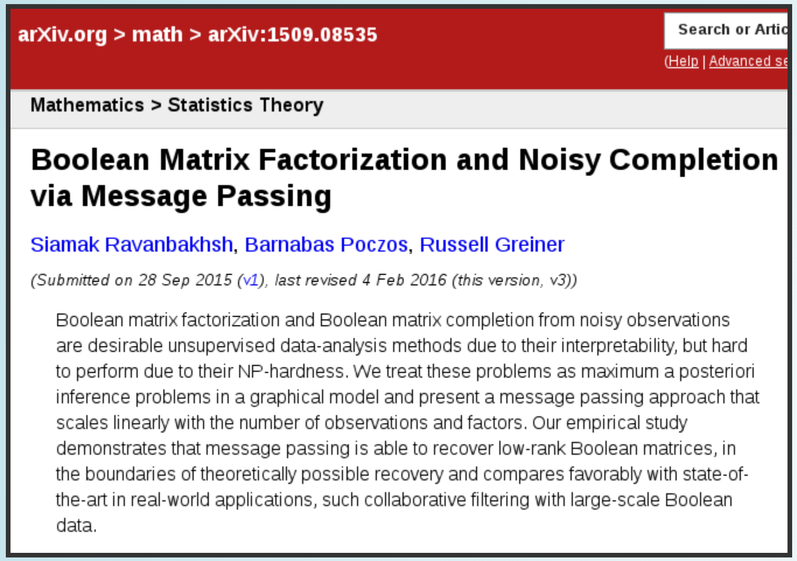
MOVIELENSE

	OBSERVED PERCENT. OF AVAILABLE RATINGS					
	1%	5%	10%	20%	50%	95%
1M						
ORM	63.4	67.0	68.5	69.8	70.9	71.2
MP	56.7	64.9	67.2	68.8	70.7	71.5
DEEP ORM	63.8	67.2	68.6	70.0	71.4	72.1
100K						
ORM	57.3	62.6	64.4	66.3	68.6	70.0
MP	52.8	60.7	63.0	65.2	67.5	69.5
DEEP ORM	58.5	63.5	65.2	66.5	68.8	70.1

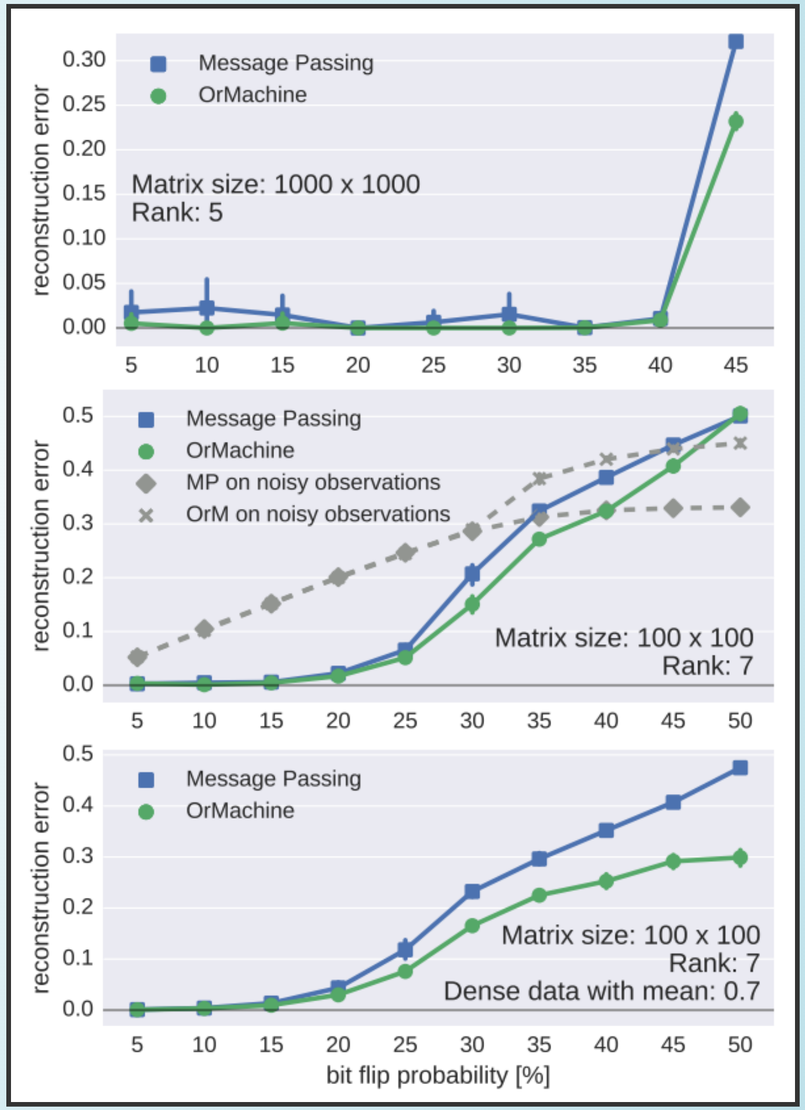
Percentages of correctly predicted, unobserved movie ratings.



RANDOM MATRIX FACTORISATION



- MAP inference using message passing.
- Outperforms all previous state-of-the-art methods.
- **OrMachine features consistently lower reconstruction error**



RANDOM MATRIX COMPLETION

- Missing dat? Set unobserved data-point to $x_{nd} = 0.5 \;\rightarrow\; \tilde{x}_{nd} = 0$

$$L = \prod_{nd} \sigma \left[\lambda \tilde{x}_{nd} (1 - 2 \prod_{l} (1 - z_{nl} u_{ld}) \right] \;\rightarrow\; \text{Contribute constant factor } \sigma(0) = \frac{1}{2}$$

$$p(z_{nl} | \text{rest}) = \sigma \left[\lambda \tilde{z}_{nl} \sum_{d} \tilde{x}_{nd} \; u_{ld} \prod_{l' \neq l} (1 - z_{nl'} u_{l'd}) \right] \;\rightarrow\; \text{No contribution}$$