d-VMP: Distributed Variational Message Passing

Andrés R. Masegosa[1], Ana M. Martínez[2],
Helge Langseth[1], Thomas D. Nielsen[2], Antonio Salmerón[3],
Darío Ramos-López[3], Anders L. Madsen[2,4]

[1]Department of Computer Science, Aalborg University, Denmark

[2] Department of Computer and Information Science,
The Norwegian University of Science and Technology, Norway

[3]Department of Mathematics, University of Almería, Spain

[4] Hugin Expert A/S, Aalborg, Denmark

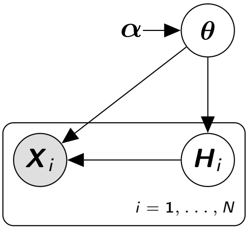
Outline

1. Motivation
2. Variational Message Passing
3. d-VMP
4. Experimental results
5. Conclusions

Outline

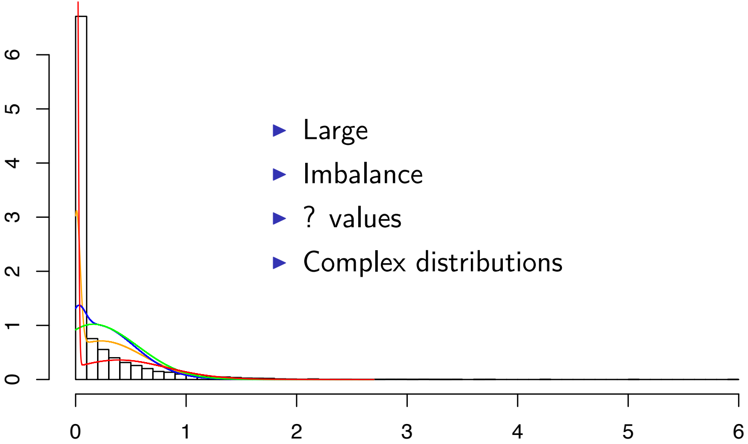
Motivation

- Goal: learn a generative model for a finantial dataset to monitor the customers and make predictions for a single customer.



Motivation

- Large
- Imbalance
- ? values
- Complex distributions

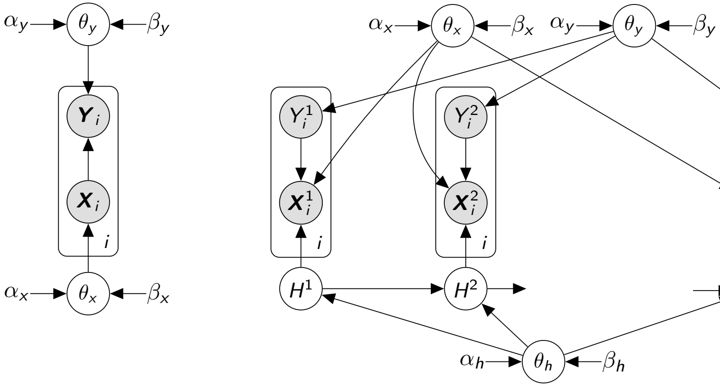


- Large
- Imbalance
- ? values
- Complex distributions

Popular existing approach: SVI

- **Stochastic Variational Inference:** iteratively updates the model parameters based on subsampled data batches.
  - No estimation of all local hidden variables of the model.
  - No generation of lower bound.
  - Poor fit if batch of data is not representative from all data.

Example of restricted models for SVI:



(a) Linear regression (b) Dynamic model

Our contribution:

- **d-VMP:** a distributed message passing scheme.
  - Defined for a broader class of models (than SVI).
  - Better and faster convergence results compared to SVI.
  - Posterior over all latent variables and the lower bound available.

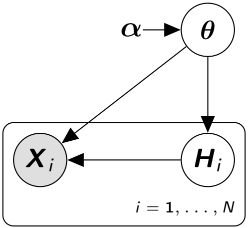
Outline

1. Motivation
2. Variational Message Passing
3. d-VMP
4. Experimental results
5. Conclusions

Models:

- Bayesian learning on iid. data using conjugate exponential BN models:

$$\ln p(X) \quad = \quad \ln h_X + \boldsymbol{s}_X \cdot \boldsymbol{\eta} - A_X(\boldsymbol{\eta})$$



Variational Inference:

- Approximate $p(\boldsymbol{\theta}, \boldsymbol{H}|\mathcal{D})$ (often intractable) by finding tractable posterior distributions $q \in \mathcal{Q}$ by minimizing:

$$\min_{q(\boldsymbol{\theta},\boldsymbol{H})\in\mathcal{Q}} KL(q(\boldsymbol{\theta}, \boldsymbol{H})|p(\boldsymbol{\theta}, \boldsymbol{H}|\mathcal{D})),$$

- In the *mean field variational* approach, $\mathcal{Q}$ is assumed to fully factorize:

$$q(\boldsymbol{\theta}, \boldsymbol{H}) = \prod_{k=1}^{M} q(\theta_k) \prod_{i=1}^{N} \prod_{j=1}^{J} q(H_{i,j}),$$

Variational Inference:

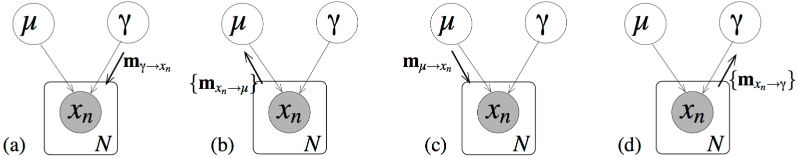
- Variational Inference exploits:

$$\underbrace{\boxed{\ln P(\mathcal{D})}}_{\text{constant}} = \underbrace{\boxed{\mathcal{L}(q(\boldsymbol{\theta}, \boldsymbol{H}))}}_{\text{Maximize}} + \underbrace{\boxed{KL(q(\boldsymbol{\theta}, \boldsymbol{H})|p(\boldsymbol{\theta}, \boldsymbol{H}|\mathcal{D}))}}_{\text{Minimize}},$$

- Iterative coordinate ascent of the variational distributions.
- Updates in the variational distribution of a variable only involves variables in its Markov blanket.
- Coordinate ascent algorithm formulated as a message passing scheme.

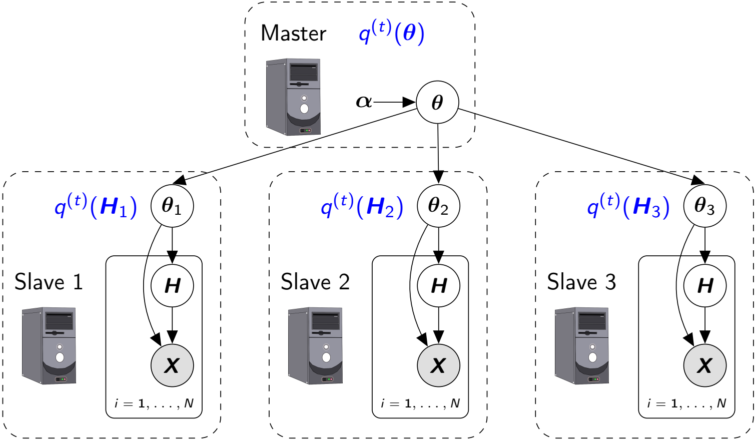
Variational Message Passing, VMP:

- **Message from parent to child:** moment parameters (expectation of the sufficient statistics).
- **Message from child to parent:** natural parameters (based on the messages received from the co-parents).



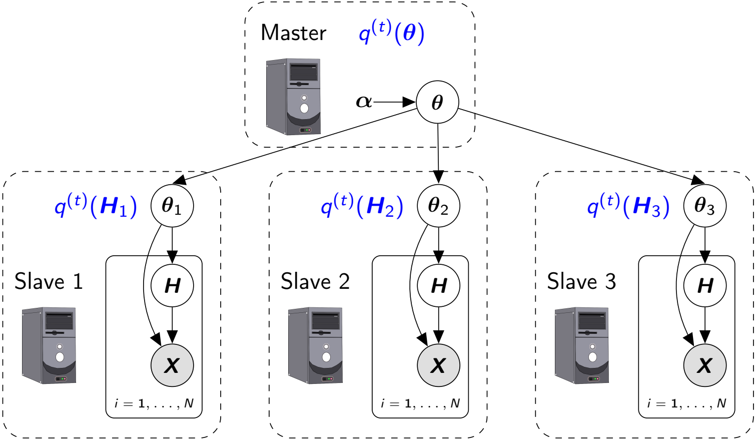
Outline

Distributed optimization of the lower bound:



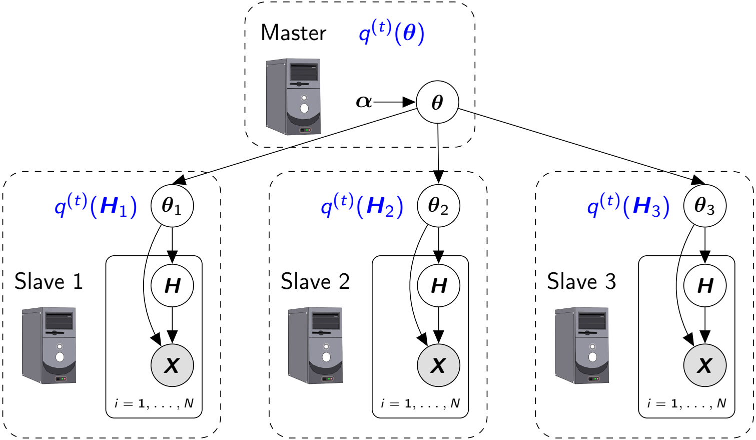
$q^{(t)}(\boldsymbol{\theta})$ is **broadcasted** to all the slave nodes.

Distributed optimization of the lower bound:



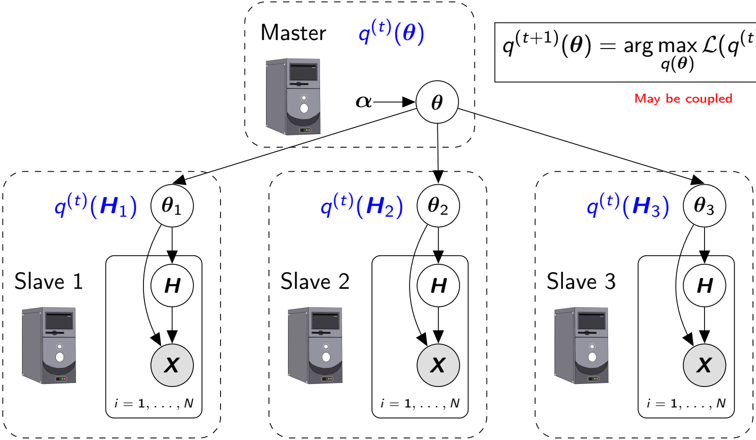
$$q^{(t+1)}(\boldsymbol{H}) = \arg\max_{q(\boldsymbol{H})} \mathcal{L}(q(\boldsymbol{H}), q^{(t)}(\boldsymbol{\theta}))$$

Distributed optimization of the lower bound:



$$q^{(t+1)}(\boldsymbol{H}_n) = \arg\max_{q(\boldsymbol{H}_n)} \mathcal{L}_n(q(\boldsymbol{H}_n), q^{(t)}(\boldsymbol{\theta}))$$

Distributed optimization of the lower bound:



$$\boxed{q^{(t+1)}(\boldsymbol{\theta}) = \arg\max_{q(\boldsymbol{\theta})} \mathcal{L}(q^{(t)}(\boldsymbol{H}), q(\boldsymbol{\theta}))}$$

May be coupled

Candidate solutions:

- Resort to a generalized mean-field approximation as SVI: does not factorize over the global parameters.
  - Prohibitive for models with a large number of global (coupled) parameters, e.g. linear regression.
- Our proposal: **VMP as a distributed projected natural gradient ascent algorithm (PGNA)**.

d-VMP as a projected natural gradient ascent

- **Insight 1:** VMP can be expressed as a projected natural gradient ascent algorithm.

$$\boldsymbol{\eta}_X^{(t+1)} = \boldsymbol{\eta}_X^{(t)} + \rho_{X,t}[\hat{\nabla}_{\boldsymbol{\eta}}\mathcal{L}(\boldsymbol{\eta}^{(t)})]_X^{+} \quad (1)$$

  - $[\cdot]$ is the projection operator.

d-VMP as a projected natural gradient ascent

- **Insight 2:** The *natural gradient* of the lower bound can be expressed as follows:

$$\hat{\nabla}_{\boldsymbol{\eta}_{\boldsymbol{\theta}}}\mathcal{L} = \mathbf{m}_{Pa(\boldsymbol{\theta})\rightarrow\boldsymbol{\theta}} + \sum \mathbf{m}_{H_i\rightarrow\boldsymbol{\theta}}$$

  - The gradient can be computed in parallel.

d-VMP as a projected natural gradient ascent

- **Insight 3:** Global parameters are "coupled" only if they belong to each other's Markov blanket.
  - Define a disjoint partition of the global parameters:

$$\mathcal{R} = \{\mathcal{J}_1, \ldots, \mathcal{J}_S\}$$

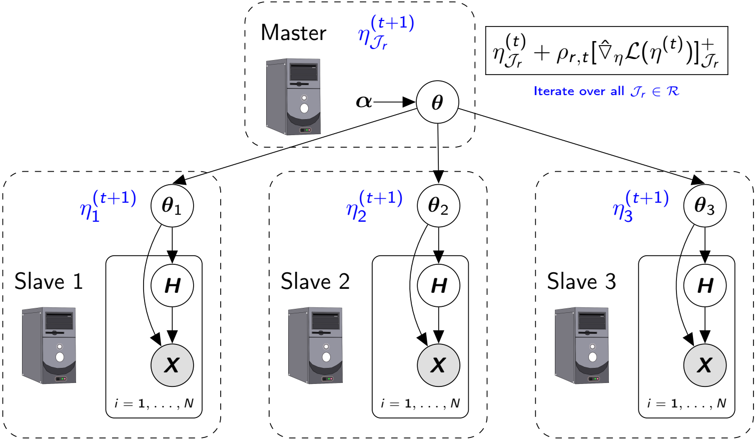
d-VMP as a projected natural gradient ascent

- d-VMP is based on performing independent global updates over the global parameters of each partition:

$$\boldsymbol{\eta}_{\mathcal{J}_r}^{(t+1)} \quad = \quad \boldsymbol{\eta}_{\mathcal{J}_r}^{(t)} + \rho_{r,t}[\hat{\nabla}_{\boldsymbol{\eta}}\mathcal{L}(\boldsymbol{\eta}^{(t)})]_{\mathcal{J}_r}^{+}$$

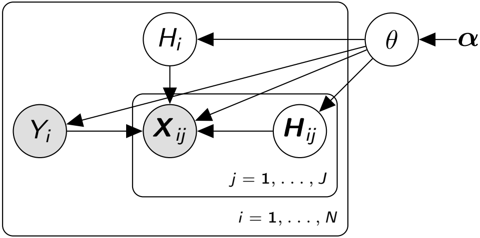
  - $\rho_{r,t}$ is the learning rate. If $|\mathcal{J}_r| = 1$ then $\rho_{r,t} = 1$.

dVMP as a distributed PNGA algorithm:



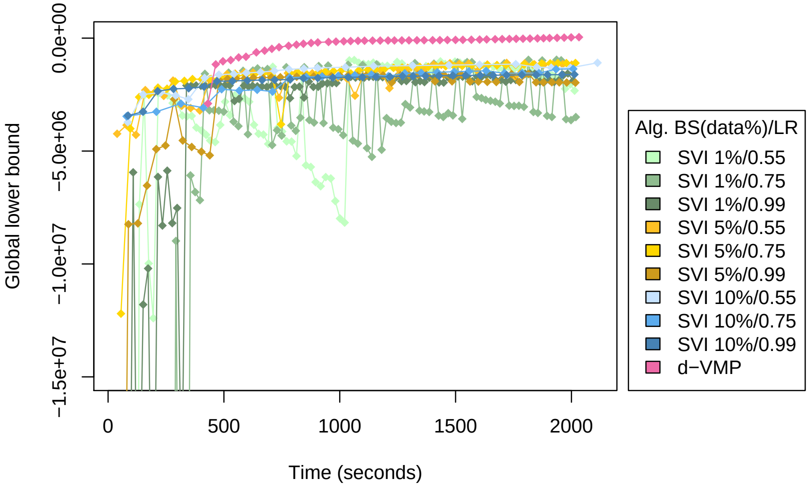
Outline

Model fit to the data



- Representative sample of 55K clients (N) and 33 attributes (J).
- "Unrolled" model of more than **3.5M nodes** (75% latent variables).

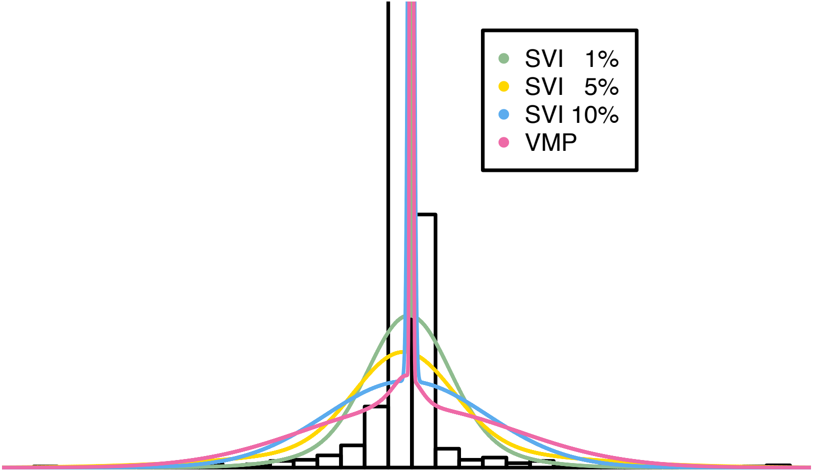
Model fit to the data



Test marginal log-likelihood

	BS (% data)	LR	Log-Likel.
SVI	1 %	0.55	-180902.87
		0.75	-298564.03
		0.99	-426979.52
	5 %	0.55	-177302.24
		0.75	-333264.16
		0.99	-628105.70
	10 %	0.55	-347035.22
		0.75	-397525.45
		0.99	-538087.13
d-VMP		1.0	67265.34

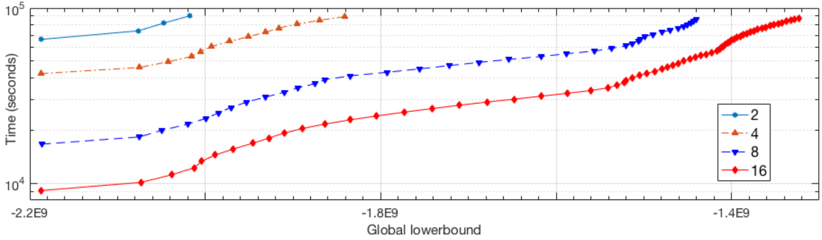
Mixtures of learnt posteriors for one attribute



Scalability settings

- Generated data set of 42 million samples per client and 12 variables.
- "Unrolled" model of more than **1 billion (10^9) nodes** (75% latent variables).
- AMIDST Toolbox with Apache Flink.
- Amazon Web Services (AWS) as distributed computing environment.

Scalability results



Outline

1. Motivation
2. Variational Message Passing
3. d-VMP
4. Experimental results
5. Conclusions

Conclusions

- Variational methods can be scaled using distributed computation instead of sampling techniques.

- Bayesian learning in model with more than 1 billion nodes (75% of hidden).

Thank you for your attention

Questions?

You can download our open source Java toolbox:
`amidsttoolbox.com`

Acknowledgments: This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619209