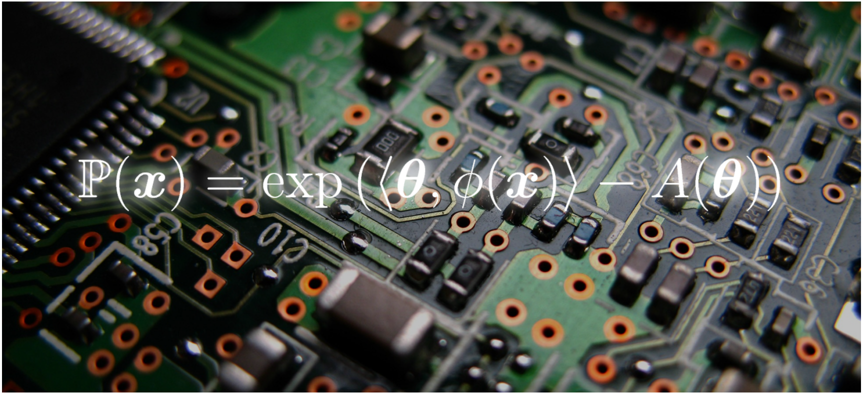
technische universität
dortmund



Exponential Families on Resource-Constrained Systems

Bayes Forum, May 4, 2018

Nico Piatkowski

Artificial Intelligence Group

My favorite co-authors at SFB876

Prof. Dr. Katharina Morik

Dr. Sangkyun Lee

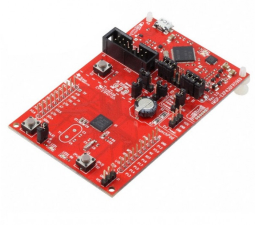
Sibylle Hess

Funded by DFG via SFB876: “Providing Information by Resource-Constrained Data Analysis”

SFB 876 Verfügbarkeit von Information durch Analyse unter Ressourcenbeschränkung

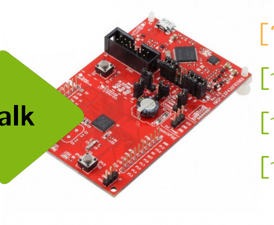
Learning on resource-constrained systems

	Cluster	Ultra-Low-Power
Feasible:	[↑]	[?]
Energy:	[↓]	[↑]
Communication:	[↓]	[↑]
Privacy:	[↓]	[↑]



Learning on resource-constrained systems

	Cluster	Ultra-Low-Power
Feasible:	[↑]	[?]
Energy:	[↓]	[↑]
Communication:	[↓]	[↑]
Privacy:	[↓]	[↑]



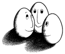
This talk

This talk

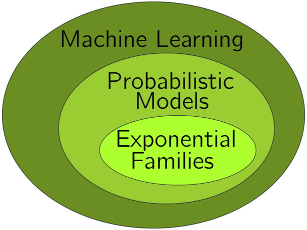
Tasks:

Reduce
- Parameter/Memory complexity
- Arithmetic complexity
- Computational complexity

with guarantees!



Exponential Families



Memory/Computation — Sufficient statistic

Computation — Normalization

Random variable

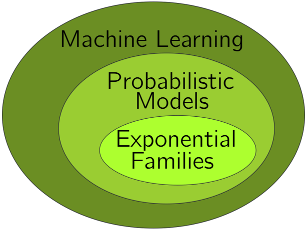
$$p_{\boldsymbol{\theta}}(\mathbf{X} = \mathbf{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A(\boldsymbol{\theta}))$$

Base — Arithmetic

Parameter — Memory

Data

Exponential Families



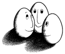
Memory/Computation — Sufficient statistic; Computation — Normalization; Random variable

$$p_{\boldsymbol{\theta}}(\mathbf{X}=\mathbf{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x})\rangle - A(\boldsymbol{\theta}))$$

Base (Arithmetic); Parameter (Memory); Data

- **Flexible**:

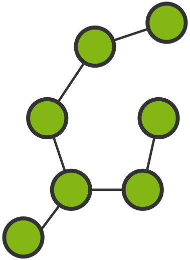
- **Unique**: Aggregation of data set $\mathcal{D}$ independent of $|\mathcal{D}|$ **iff** $p_{\boldsymbol{\theta}}$ belongs to a (generative) exponential family [Pitman/1936a].



Exponential Families as Graphical Models

Let $G = (V, E)$ encode the conditional independence structure of $\mathbf{X}$.

$$\underbrace{\frac{1}{Z(\boldsymbol{\theta})} \prod_{C \in \mathcal{C}} \exp(\langle \boldsymbol{\theta}_C, \phi_C(\mathbf{x}_C) \rangle)}_{\text{Factorization over cliques}} = \underbrace{\exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A(\boldsymbol{\theta}))}_{\text{Exponential family}}$$



Normalization

$$A(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta}) = \log \int_{\mathcal{X}} \exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle) \, \mathrm{d}\, \nu(\mathbf{x})$$

is **#P**-complete (worst-case over G!).

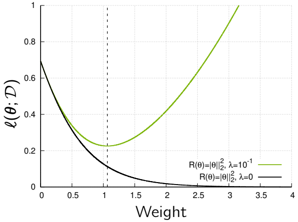
For trees in **FP** $\Rightarrow$ Variational approximations: Simplify G.

Regularized Learning

$$\ell(\boldsymbol{\theta};\mathcal{D}) = \underbrace{-\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}}(\langle\boldsymbol{\theta},\phi(\mathbf{x})\rangle - A(\boldsymbol{\theta}))}_{\text{Negative avg. log-likelihood}} + \underbrace{\lambda R(\boldsymbol{\theta})}_{\text{Regularization}}$$

Regularization: "give preference to a particular solution with desirable properties"

- Solve ill-posed problems
- Avoid overfitting
- Select relevant (groups of) features

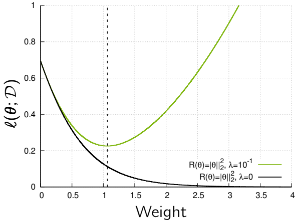


Regularized Learning

$$\ell(\boldsymbol{\theta}; \mathcal{D}) = \underbrace{-\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} (\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A(\boldsymbol{\theta}))}_{\text{Negative avg. log-likelihood}} + \underbrace{\lambda R(\boldsymbol{\theta})}_{\text{Regularization}}$$

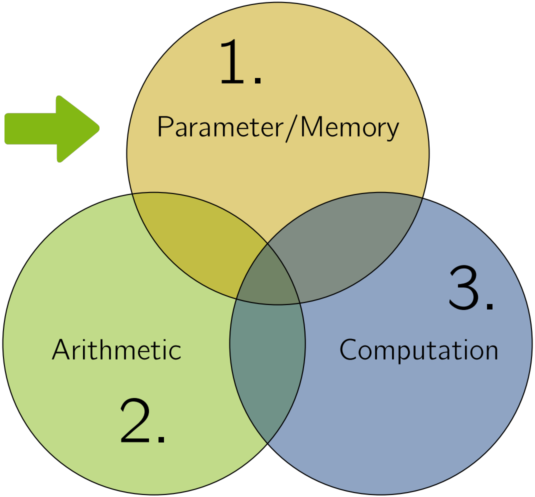
Regularization: "give preference to a particular solution with desirable properties"

- Solve ill-posed problems
- Avoid overfitting
- Select relevant (groups of) features



Here: desirable properties $\equiv$ reduced resource consumption

Reduce Resource Consumption via Regularization



1. Reduce Parameter/Memory Complexity

Main influencing factor: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{|\mathcal{C}|})$ (d-dimensional)

Motivation: Physics [Ising/1925] and natural language processing [Lafferty/etal/2001]:

- Reparametrization: $\boldsymbol{\theta}$ is function of low-dimensional Δ
- Parameter sharing: Multiple cliques share the same $\boldsymbol{\theta}_C$

Problem: Domain specific (Ferromagnetism/Language model)

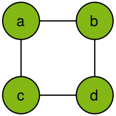
Task: Find generic reparametrization/parameter sharing

Temporal Models

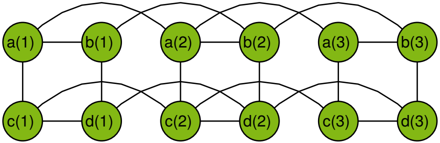
Resource-constrained devices collect data over time

Multivariate time series: $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T)$, $\mathbf{X}_i \in \mathcal{Q}^n$

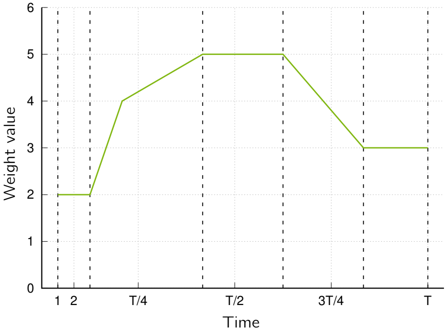
Time-dependent weights: $\boldsymbol{\theta}_C(t)$



Spatial graph



T temporal copies (interconnected)

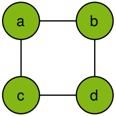


Temporal Models

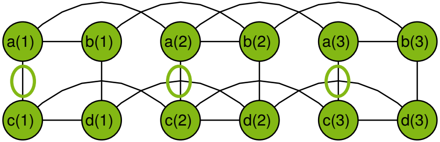
Resource-constrained devices collect data over time

Multivariate time series: $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T)$, $\mathbf{X}_i \in \mathcal{Q}^n$

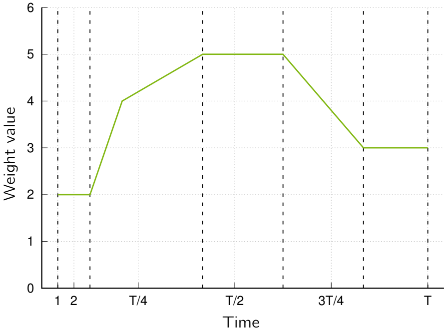
Time-dependent weights: $\boldsymbol{\theta}_C(t)$



Spatial graph

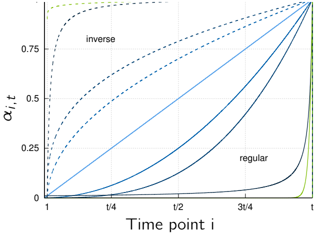


T temporal copies (interconnected)



Reduction via Regularized Reparametrization

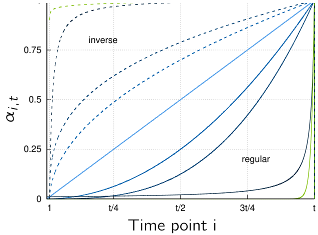
$$\boldsymbol{\theta}_C(t) = \sum_{i=1}^{t} \alpha_{i,t} \underbrace{\boldsymbol{\Delta}_C(i)}_{\text{New learnable parameters}}$$



with ($\alpha_{t,t} = 1$). Coefficients $\alpha_{i,t}$ control influence of previous time points on $\boldsymbol{\theta}_C(t)$.

Reduction via Regularized Reparametrization

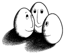
$$\boldsymbol{\theta}_C(t) = \sum_{i=1}^{t} \alpha_{i,t} \underbrace{\boldsymbol{\Delta}_C(i)}_{\text{New learnable parameters}}$$



with $(\alpha_{t,t} = 1)$. Coefficients $\alpha_{i,t}$ control influence of previous time points on $\boldsymbol{\theta}_C(t)$.

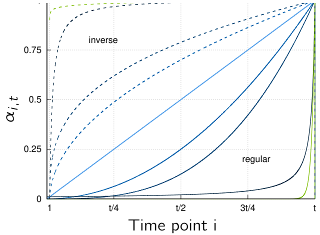
$$\ell(\boldsymbol{\Delta}; \mathcal{D}) = \underbrace{-\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} (\langle \boldsymbol{\theta}(\boldsymbol{\Delta}), \phi(\mathbf{x}) \rangle - A(\boldsymbol{\theta}(\boldsymbol{\Delta})))}_{\text{Negative avg. log-likelihood}} + \underbrace{\lambda_1 \|\boldsymbol{\Delta}\|_1 + \frac{\lambda_2}{2} \|\boldsymbol{\Delta}\|_2^2}_{\text{Regularization}}$$

[Piatkowski/etal/2013] (Machine Learning Journal; Best student paper at ECML-PKDD 2013)
[Piatkowski/Schnitzler/2016]



Reduction via Regularized Reparametrization

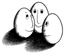
$$\boldsymbol{\theta}_C(t) = \sum_{i=1}^{t} \alpha_{i,t} \underbrace{\boldsymbol{\Delta}_C(i)}_{\text{New learnable parameters}}$$



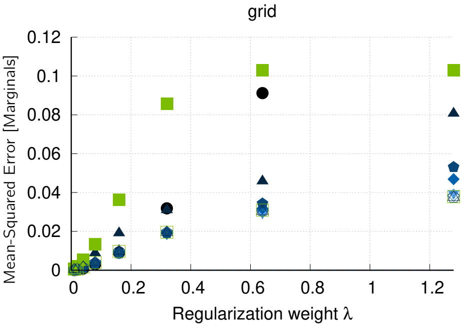
with $(\alpha_{t,t} = 1)$. Coefficients $\alpha_{i,t}$ control influence of previous time points on $\boldsymbol{\theta}_C(t)$.

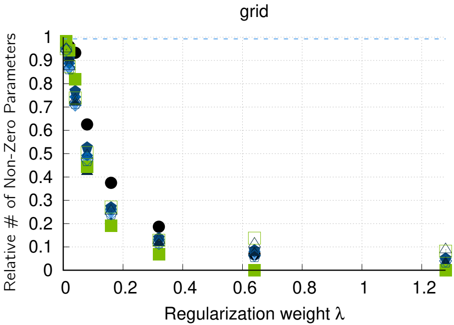
$$\ell(\boldsymbol{\Delta}; \mathcal{D}) = \underbrace{-\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} (\langle \boldsymbol{\theta}(\boldsymbol{\Delta}), \phi(\mathbf{x}) \rangle - A(\boldsymbol{\theta}(\boldsymbol{\Delta})))}_{\text{Negative avg. log-likelihood}} + \underbrace{\lambda_1 \|\boldsymbol{\Delta}\|_1 + \frac{\lambda_2}{2} \|\boldsymbol{\Delta}\|_2^2}_{\text{Regularization}}$$

[Piatkowski/etal/2013] (Machine Learning Journal; Best student paper at ECML-PKDD 2013)
[Piatkowski/Schnitzler/2016]



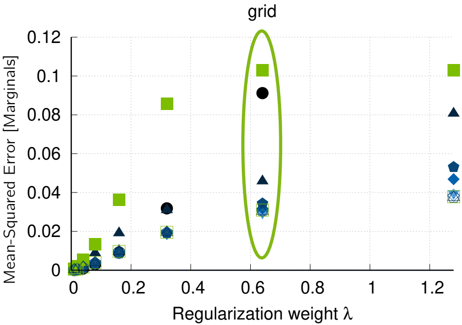
Empirical Demonstration (Synthetic Grid)

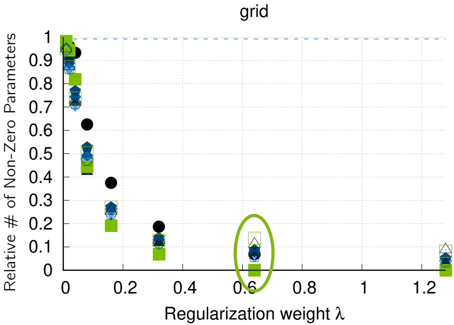




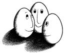
- Black circle is plain l_1-regularization
- Proposed approach achieves higher sparsity at lower error

Empirical Demonstration (Synthetic Grid)

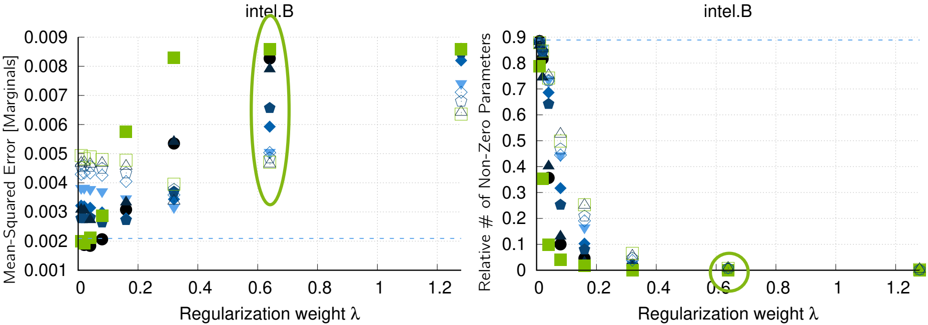




- Black circle is plain l_1-regularization
- Proposed approach achieves higher sparsity at lower error

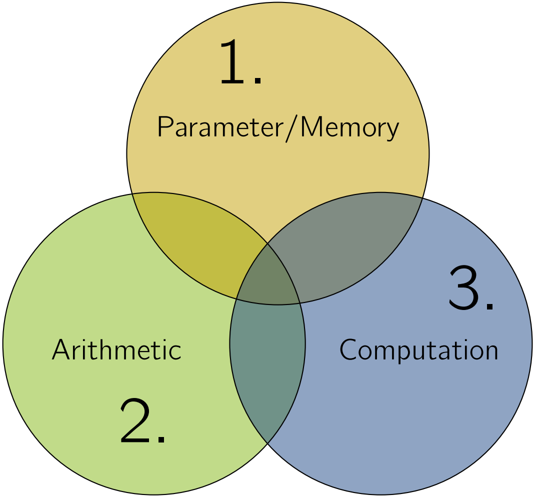


Empirical Demonstration (Intel Lab)



- Black circle is plain l_1-regularization
- Proposed approach achieves higher sparsity at lower error

Reduce Resource Consumption via Regularization



2. Reduce Arithmetic Complexity

Evaluating $\exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A(\boldsymbol{\theta}))$ requires real-valued arithmetic

Motivation: Empirical work on neural networks [Khan/Hines/1994] and Bayesian network classifiers [Tschiatschek/etal/2012]:

- Truncation: Prune fractional digits of learned parameters
- Restricted parameter set: $\boldsymbol{\theta}_i$ is constrained to a subset of `float`

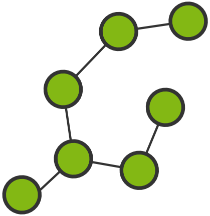
Problem: No integer-valued inference / learning procedure

Task: Formalize and devise integer learning for exponential families

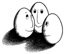
Base-2 Exponential Families

Based on a proof from [Pitman/1936a]:

$$p_{\boldsymbol{\theta}}(\mathbf{X} = \mathbf{x}) = 2^{\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A_2(\boldsymbol{\theta})}$$



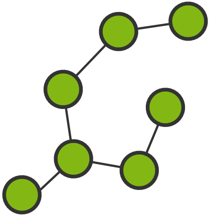
- Equivalent to base-e model.
- $\boldsymbol{\theta} \in \mathbb{N}^d \Rightarrow$ integer arithmetic suffices.



Base-2 Exponential Families

Based on a proof from [Pitman/1936a]:

$$p_{\boldsymbol{\theta}}(\mathbf{X} = \mathbf{x}) = 2^{\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A_2(\boldsymbol{\theta})}$$



- Equivalent to base-e model.
- $\boldsymbol{\theta} \in \mathbb{N}^d \Rightarrow$ integer arithmetic suffices.

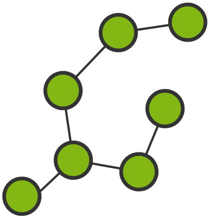
Motivated by belief propagation: **Bit-Length Propagation**

$$b_{v\to u}(x_u) = \text{bitLength} \sum_{x_v \in \mathcal{X}_v} 2^{\boldsymbol{\theta}_{(v,u)=(x_v,x_u)} + \sum_{w \in \mathcal{N}(v)\setminus\{u\}} b_{w\to v}(x_v)}$$

Kullback-Leibler divergence depends on longest path and degree.

[Piatkowski/etal/2016a] (Neurocomputing Journal; Selected paper at ICPRAM 2014)

Base-2 Exponential Families



Based on a proof from [Pitman/1936a]:

$$p_{\boldsymbol{\theta}}(\mathbf{X} = \mathbf{x}) = 2^{\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A_2(\boldsymbol{\theta})}$$

- Equivalent to base-e model.
- $\boldsymbol{\theta} \in \mathbb{N}^d \Rightarrow$ integer arithmetic suffices.

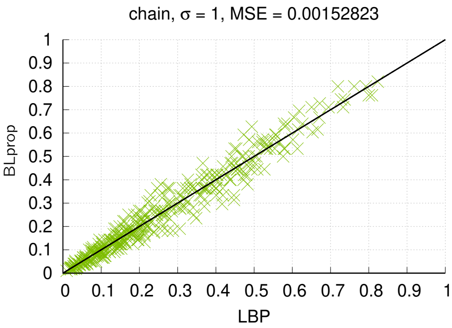
Motivated by belief propagation: **Bit-Length Propagation**

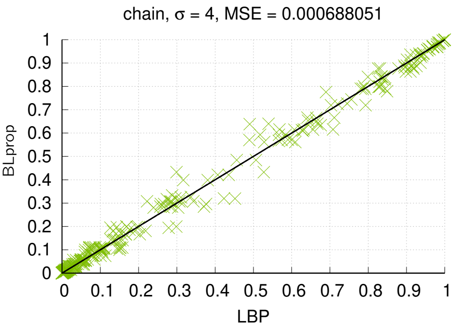
$$b_{v \to u}(x_u) = \text{bitLength} \sum_{x_v \in \mathcal{X}_v} 2^{\boldsymbol{\theta}_{(v,u)=(x_v,x_u)} + \sum_{w \in \mathcal{N}(v) \setminus \{u\}} b_{w \to v}(x_v)}$$

Kullback-Leibler divergence depends on longest path and degree.

[Piatkowski/etal/2016a] (Neurocomputing Journal; Selected paper at ICPRAM 2014)

Empirical Demonstration (Marginals)

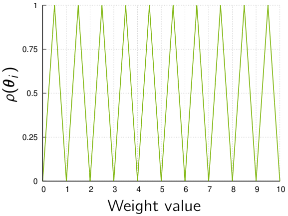




- Increased parameter variance decreases estimation error

Integer Regularization

$$\lambda R_{\texttt{int}}(\boldsymbol{\theta}) = \lambda \sum_{i=1}^{d} \underbrace{1 - |1 - 2(\lceil \boldsymbol{\theta}_i \rceil - \boldsymbol{\theta}_i)|}_{\rho(\boldsymbol{\theta}_i)}$$



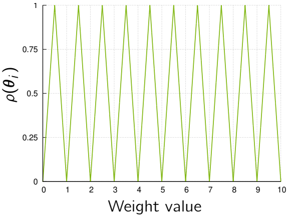
Non-smooth non-convex minimization via proximal method [Bolte/etal/2014]: $\boldsymbol{\theta}^{(j+1)} = \operatorname{prox}_{\lambda R_{\texttt{int}}}(\boldsymbol{\theta}^{(j)} + \eta \nabla \ell(\boldsymbol{\theta}; \mathcal{D}))$

$$\operatorname{prox}_{\lambda R_{\texttt{int}}}(\boldsymbol{\theta})_i := \begin{cases} \operatorname{round}(\boldsymbol{\theta}_i) & \text{, if } |\omega - \boldsymbol{\theta}_i| \leq 2\lambda \\ \boldsymbol{\theta}_i + 2\lambda & \text{, else if } \omega > \boldsymbol{\theta}_i \\ \boldsymbol{\theta}_i - 2\lambda & \text{, else if } \omega < \boldsymbol{\theta}_i \end{cases}$$

with $\omega := \arg\min_{u \in \mathbb{N}} |u - \boldsymbol{\theta}_i|$. $\lambda \geq 1/4$ ensures integrality!

Integer Regularization

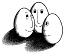
$$\lambda R_{\texttt{int}}(\boldsymbol{\theta}) = \lambda \sum_{i=1}^{d} \underbrace{1 - |1 - 2(\lceil \boldsymbol{\theta}_i \rceil - \boldsymbol{\theta}_i)|}_{\rho(\boldsymbol{\theta}_i)}$$



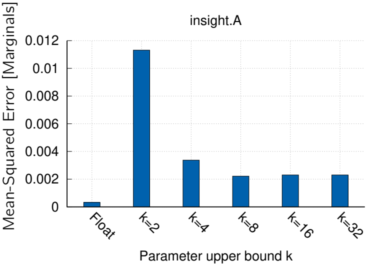
Non-smooth non-convex minimization via proximal method [Bolte/etal/2014]: $\boldsymbol{\theta}^{(j+1)} = \text{prox}_{\lambda R_{\texttt{int}}}(\boldsymbol{\theta}^{(j)} + \eta \nabla \ell(\boldsymbol{\theta}; \mathcal{D}))$

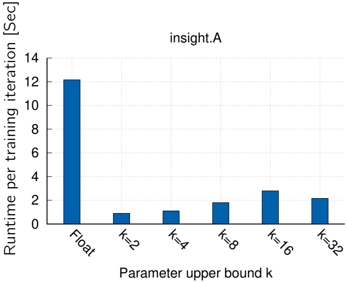
$$\text{prox}_{\lambda R_{\texttt{int}}}(\boldsymbol{\theta})_i := \begin{cases} \text{round}(\boldsymbol{\theta}_i) & \text{, if } |\omega - \boldsymbol{\theta}_i| \leq 2\lambda \\ \boldsymbol{\theta}_i + 2\lambda & \text{, else if } \omega > \boldsymbol{\theta}_i \\ \boldsymbol{\theta}_i - 2\lambda & \text{, else if } \omega < \boldsymbol{\theta}_i \end{cases}$$

with $\omega := \arg\min_{u \in \mathbb{N}} |u - \boldsymbol{\theta}_i|$. $\lambda \geq 1/4$ ensures integrality!

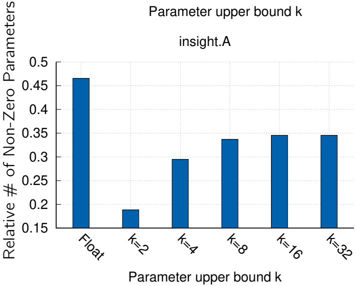


Empirical Demonstration (Learning)

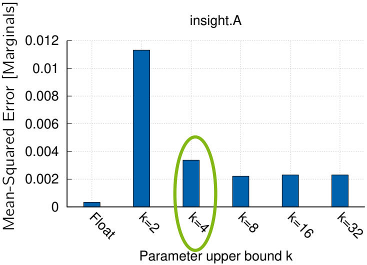


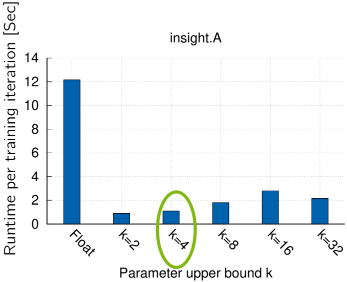


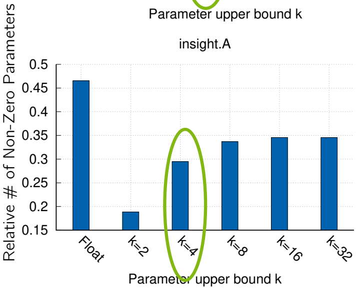
- Maintain low error while achieving $\approx 10\times$ speed-up on cluster hardware.



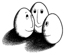
Empirical Demonstration (Learning)



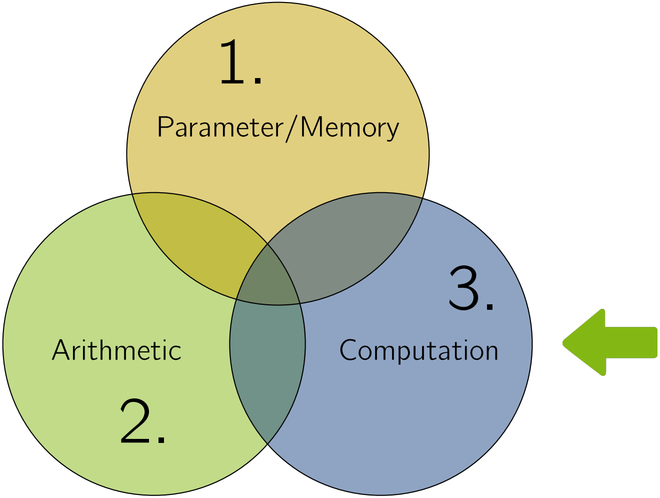




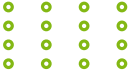
- Maintain low error while achieving $\approx 10\times$ speed-up on cluster hardware.



Reduce Resource Consumption via Regularization



3. Reduce Computational Complexity


Naive MF

Evaluating $A(\boldsymbol{\theta})$ is **#P**-complete

Motivation: Variational inference [Wainwright/Jordan/2008] and discrete integration by hashing [Ermon/2013]:

- Variational: Minimize KL to "simpler" surrogate
- WISH: Randomized Riemann sum approximation to $Z(\boldsymbol{\theta})$

Problem 1: No error bounds for simplification.
Problem 2: Tight bounds for discrete integration but still **NP**-hard.

Task: Find a way to trade quality against complexity.

Quadrature

Based on numerical integration [Clenshaw/Curtis/1960]:

$$I[f] = \int f(x)\,\mathrm{d}\,x \approx \int \hat{f}(x)\,\mathrm{d}\,x = \hat{I}[f]$$

Error is bounded when $\|\hat{f} - f\|_\infty$ is upper bounded.

[Piatkowski/Morik/2016] (ICML 2016)

Quadrature

Based on numerical integration [Clenshaw/Curtis/1960]:

$$I[f] = \int f(x)\,\mathrm{d}\,x \approx \int \hat{f}(x)\,\mathrm{d}\,x = \hat{I}[f]$$

Error is bounded when $\|\hat{f} - f\|_\infty$ is upper bounded.

Degree-k polynomial approximation $\hat{f}_k$ for $f(\mathbf{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x})\rangle)$:

$$\hat{Z}_k(\boldsymbol{\theta}) = \int_{\mathcal{X}} \hat{f}_k(\mathbf{x})\,\mathrm{d}\,\nu(\mathbf{x}) = \sum_{i=0}^{i} \mathbf{c}_i \sum_{\mathbf{j}\in[d]^i} \prod_{l=1}^{i} \boldsymbol{\theta}_{\mathbf{j}(l)} \underbrace{\int_{\mathcal{X}} \prod_{l=1}^{i} \phi_{\mathbf{j}(l)}\,\mathrm{d}\,\nu(\mathbf{x})}_{\text{Independent of } \boldsymbol{\theta}}$$

[Piatkowski/Morik/2016] (ICML 2016)

Quadrature

Based on numerical integration [Clenshaw/Curtis/1960]:

$$I[f] = \int f(x)\,\mathrm{d}\,x \approx \int \hat{f}(x)\,\mathrm{d}\,x = \hat{I}[f]$$

Error is bounded when $\|\hat{f} - f\|_\infty$ is upper bounded.

Degree-k polynomial approximation $\hat{f}_k$ for $f(\mathbf{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x})\rangle)$:

$$\hat{Z}_k(\boldsymbol{\theta}) = \int_{\mathcal{X}} \hat{f}_k(\mathbf{x})\,\mathrm{d}\,\nu(\mathbf{x}) = \sum_{i=0}^{i} \mathbf{c}_i \sum_{\mathbf{j}\in[d]^i} \prod_{l=1}^{i} \boldsymbol{\theta}_{\mathbf{j}(l)} \underbrace{\int_{\mathcal{X}} \prod_{l=1}^{i} \phi_{\mathbf{j}(l)}\,\mathrm{d}\,\nu(\mathbf{x})}_{\text{Independent of } \boldsymbol{\theta}}$$

[Piatkowski/Morik/2016] (ICML 2016)

Quadrature

Based on numerical integration [Clenshaw/Curtis/1960]:

$$I[f] = \int f(x)\,\mathrm{d}\,x \approx \int \hat{f}(x)\,\mathrm{d}\,x = \hat{I}[f]$$

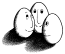
Error is bounded when $\|\hat{f} - f\|_\infty$ is upper bounded.

Degree-k polynomial approximation $\hat{f}_k$ for $f(\mathbf{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle)$:

$$\hat{Z}_k(\boldsymbol{\theta}) = \int_{\mathcal{X}} \hat{f}_k(\mathbf{x})\,\mathrm{d}\,\nu(\mathbf{x}) = \sum_{i=0}^{i} \mathbf{c}_i \sum_{\mathbf{j} \in [d]^i} \prod_{l=1}^{i} \boldsymbol{\theta}_{\mathbf{j}(l)} \chi^i(\mathbf{j})$$

[Piatkowski/Morik/2016] (ICML 2016)

Closed-form of $\chi^i(\mathbf{j})$ for various models!



Quadrature

Based on numerical integration [Clenshaw/Curtis/1960]:

$$I[f] = \int f(x)\,\mathrm{d}\,x \approx \int \hat{f}(x)\,\mathrm{d}\,x = \hat{I}[f]$$

Error is bounded when $\|\hat{f} - f\|_\infty$ is upper bounded.

Degree-k polynomial approximation $\hat{f}_k$ for $f(\mathbf{x}) = \exp(\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle)$:

$$\hat{Z}_k(\boldsymbol{\theta}) = \int_{\mathcal{X}} \hat{f}_k(\mathbf{x})\,\mathrm{d}\,\nu(\mathbf{x}) = \sum_{i=0}^{i} \mathbf{c}_i \sum_{\mathbf{j}\in[d]^i} \prod_{l=1}^{i} \boldsymbol{\theta}_{\mathbf{j}(l)} \chi^i(\mathbf{j})$$

[Piatkowski/Morik/2016] (ICML 2016)

Closed-form of $\chi^i(\mathbf{j})$ for various models!

Randomization

Enumerating $[d]^i$ in **FP** but still expensive for large d, i.
Define random variables I and $\mathbf{J}$ with

$$\mathbb{P}_{\mathbf{c}}(I = i) = \frac{|\mathbf{c}_i| \|\chi^i\|_1}{\tau} \qquad \mathbb{P}(\mathbf{J} = \mathbf{j} \mid I = i) = \frac{\chi^i(\mathbf{j})}{\|\chi^i\|_1}$$

with $\tau = \sum_{j=0}^{k} |\mathbf{c}_j| \|\chi^j\|_1$ and $\|\chi^i\|_1 = \sum_{\mathbf{j} \in [d]^i} |\chi^i(\mathbf{j})|$. Then

$$\mathbb{E}_{I,\mathbf{J}} \left[\tau \operatorname{sgn}(\mathbf{c}_I) \prod_{r=0}^{I} \boldsymbol{\theta}_{\mathbf{J}_r} \right] = \hat{Z}_k(\boldsymbol{\theta})$$

Sampling I and $\mathbf{J}$ $\Rightarrow$ Monte Carlo algorithm for $\hat{Z}_k(\boldsymbol{\theta})$

Randomization

Enumerating $[d]^i$ in **FP** but still expensive for large d, i.
Define random variables I and $\mathbf{J}$ with

$$\mathbb{P}_{\mathbf{c}}(I = i) = \frac{|\mathbf{c}_i| \|\chi^i\|_1}{\tau} \qquad \mathbb{P}(\mathbf{J} = \mathbf{j} \mid I = i) = \frac{\chi^i(\mathbf{j})}{\|\chi^i\|_1}$$

with $\tau = \sum_{j=0}^{k} |\mathbf{c}_j| \|\chi^j\|_1$ and $\|\chi^i\|_1 = \sum_{\mathbf{j} \in [d]^i} |\chi^i(\mathbf{j})|$. Then

$$\mathbb{E}_{I,\mathbf{J}} \left[\tau \operatorname{sgn}(\mathbf{c}_I) \prod_{r=0}^{I} \boldsymbol{\theta}_{\mathbf{J}_r} \right] = \hat{Z}_k(\boldsymbol{\theta})$$

Sampling I and $\mathbf{J}$ $\Rightarrow$ Monte Carlo algorithm for $\hat{Z}_k(\boldsymbol{\theta})$
$\Rightarrow$ Approximation to $Z(\boldsymbol{\theta})$ via error bound on $\hat{f}_k$

Randomization

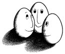
Enumerating $[d]^i$ in **FP** but still expensive for large d, i.
Define random variables I and $\mathbf{J}$ with

$$\mathbb{P}_{\mathbf{c}}(I = i) = \frac{|\mathbf{c}_i| \|\chi^i\|_1}{\tau} \qquad \mathbb{P}(\mathbf{J} = \mathbf{j} \mid I = i) = \frac{\chi^i(\mathbf{j})}{\|\chi^i\|_1}$$

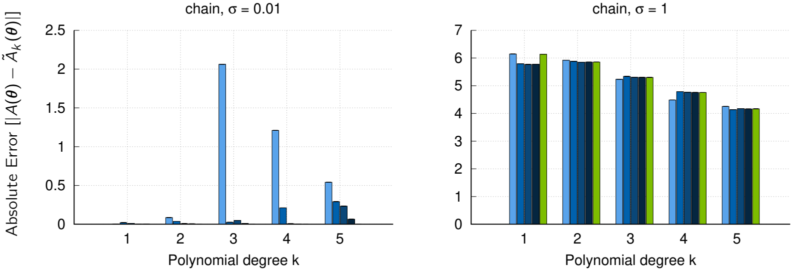
with $\tau = \sum_{j=0}^{k} |\mathbf{c}_j| \|\chi^j\|_1$ and $\|\chi^i\|_1 = \sum_{\mathbf{j} \in [d]^i} |\chi^i(\mathbf{j})|$. Then

$$\mathbb{E}_{I,\mathbf{J}} \left[\tau \, \mathrm{sgn}(\mathbf{c}_I) \prod_{r=0}^{I} \boldsymbol{\theta}_{\mathbf{J}_r} \right] = \hat{Z}_k(\boldsymbol{\theta})$$

Sampling I and $\mathbf{J}$ $\Rightarrow$ Monte Carlo algorithm for $\hat{Z}_k(\boldsymbol{\theta})$
$\Rightarrow$ Approximation to $Z(\boldsymbol{\theta})$ via error bound on $\hat{f}_k$

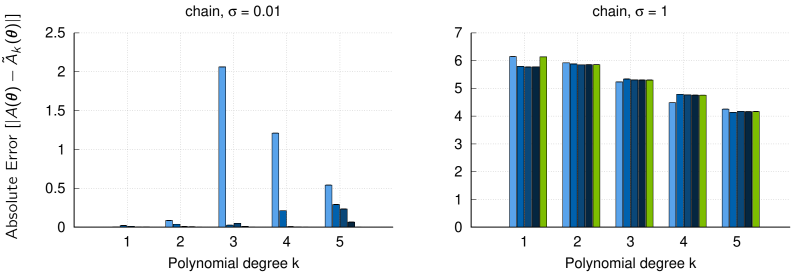


Empirical Demonstration (Log-Partition Function)



- Error decreases with increasing polynomial degree
- When $\|\boldsymbol{\theta}\|_2$ is low: Number of samples dominates error
- When $\|\boldsymbol{\theta}\|_2$ is large: Polynomial approximation dominates error

Empirical Demonstration (Log-Partition Function)



- Error decreases with increasing polynomial degree
- When $\|\boldsymbol{\theta}\|_2$ is low: Number of samples dominates error
- When $\|\boldsymbol{\theta}\|_2$ is large: Polynomial approximation dominates error

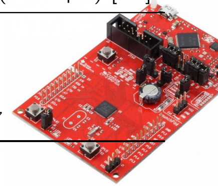
Empirical Demonstration (ULP device)

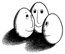
Memory:

Data	d	**None**	l_1-**Reg.**	**Reparam.** [KiB]
Chain	1066.68	4266.72	247.52	**202.08**
Star	1084.0	4336.0	**201.6**	**197.44**
Grid	1037.8	4151.2	277.92	**199.04**
Full	843.8	3375.2	257.92	**181.6**

Runtime:

Data	$\|E\|$	**LBP** (1 iter)	**BLprop** (1 iter)	**SQM** (1 sample) [ms]
Chain	15	1156.2	**19.0**	350.3
Star	15	1140.4	**19.0**	393.1
Grid	24	1838.1	**29.5**	445.3
Full	120	9642.1	**141.2**	1549.7





Conclusion

- Proposed methods
  - arose from studying the **model** perspective
  - work with **all** exponential family members

(and beyond)

  - keep the conditional independence structure **intact**

- Towards machine learning on resource-constrained systems:
  - Increase sparsity by $> \mathbf{10}\times$
  - Decrease runtime $> \mathbf{60}\times$ on ULP hardware
- **New regularization and probabilistic inference techniques** (with error bounds)