Microsoft Research

NIPS Thanks Its Sponsors

amazon.com

Microsoft Research

Google

facebook

SKYTREE
THE MACHINE LEARNING COMPANY

TWO SIGMA

United Technologies Research Center

D E Shaw & Co

xerox

YAHOO! LABS

IBM Research

TOYOTA

DRW TRADING GROUP

millionshort

criteo

PDT PARTNERS

Springer
Machine Learning Journal

Disney Research

100 Reviewer Awards

“ Up to 100 Reviewer Awards are given to reviewers who wrote exceptionally careful, thorough and useful reviews. While the work of all reviewers is essential to the conference, these award winning reviewers are to be especially thanked for the quality of their reviews”

100 Reviewer Awards:

Aasa Feragen, Abel Rodriguez, Adam Johansen, Afshin Rostamizadeh, Akshay Krishnamurthy, Alessandro Lazaric, Anand Sarwate, Andrea Vedaldi, Andreas Argyriou, Andrew Saxe , Andrew Wilson , Animashree Anandkumar , Anna Rafferty, Archana Venkataraman, Arnak Dalalyan, Arthur Choi, Benjamin Packer, Bernardino Romera Paredes, Brian McFee, Charles Isbell, Charlie Tang, Christian Machens, Christoph Sawade, Daniel Polani, David Mimno , David Wingate, David Wipf, Dhruv Batra, Emmanuel Dauce, Florent Perronnin, Gerhard Neumann, Hua Ouyang, Ilya Sutskever, Jake Bouvrie, Jaldert Rombouts, James Hensman , James Martens, Jasper Snoek, Jennifer Gillenwater, Jens Kober, Jian Peng, John Fisher, Jonathan Huang, Joseph Austerweil, Joseph Salmon, Junzhou Huang, Justin Domke, Kenji Fukumizu , Lars Buesing, Laurent Jacob, Marc Deisenroth, Marcello Restelli, Marco Cuturi, Mariya Ishteva, Mark Schmidt, Matthew Liptrot, Matthias Hein, Matus Telgarsky, Michael Hughes, Michael Mahoney, Michel Besserve, Morteza Alamgir, Neil Houlsby, Nicholas Bryan, Nick Foti, Nicolas Heess, Paolo Favaro, Pascal Poupart, Peter Bartlett, Peter Frazier, Pierre Geurts, Ping Li, Piyush Rai, Purushottam Kar , Reggis Sabbadin, Rina Foygel, Ron Meir, Ronald Ortner, Rong Ge, Rui Kuang , Sam Gershman, Scott Linderman, Scott Yih, Sebastien Gerchinovitz, Simon Lacoste-Julien, Stephan Bach, Svetlana Lazebnik, Tamara Broderick, Thomas Dietterich, Thomas Richardson, Tom Minka, Tom Walters, Tomas Werner, Tyler Neylon, Uri Shalit, Victor Lempitsky, Vikas Singh, Xaq Pitkow, Yihong Wu, Zaid Harchaoui.

Reviews and Author Rebuttals are now online

http://papers.nips.cc/

NIPS Thanks Its Sponsors

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

amazon.com

Microsoft Research

Google

facebook

SKYTREE THE MACHINE LEARNING COMPANY

TWO SIGMA

United Technologies Research Center

D E Shaw & Co

xerox

YAHOO! LABS

IBM Research

DRW TRADING GROUP

TOYOTA

millionshort

criteo

PDT PARTNERS

Springer Machine Learning Journal

Disney Research

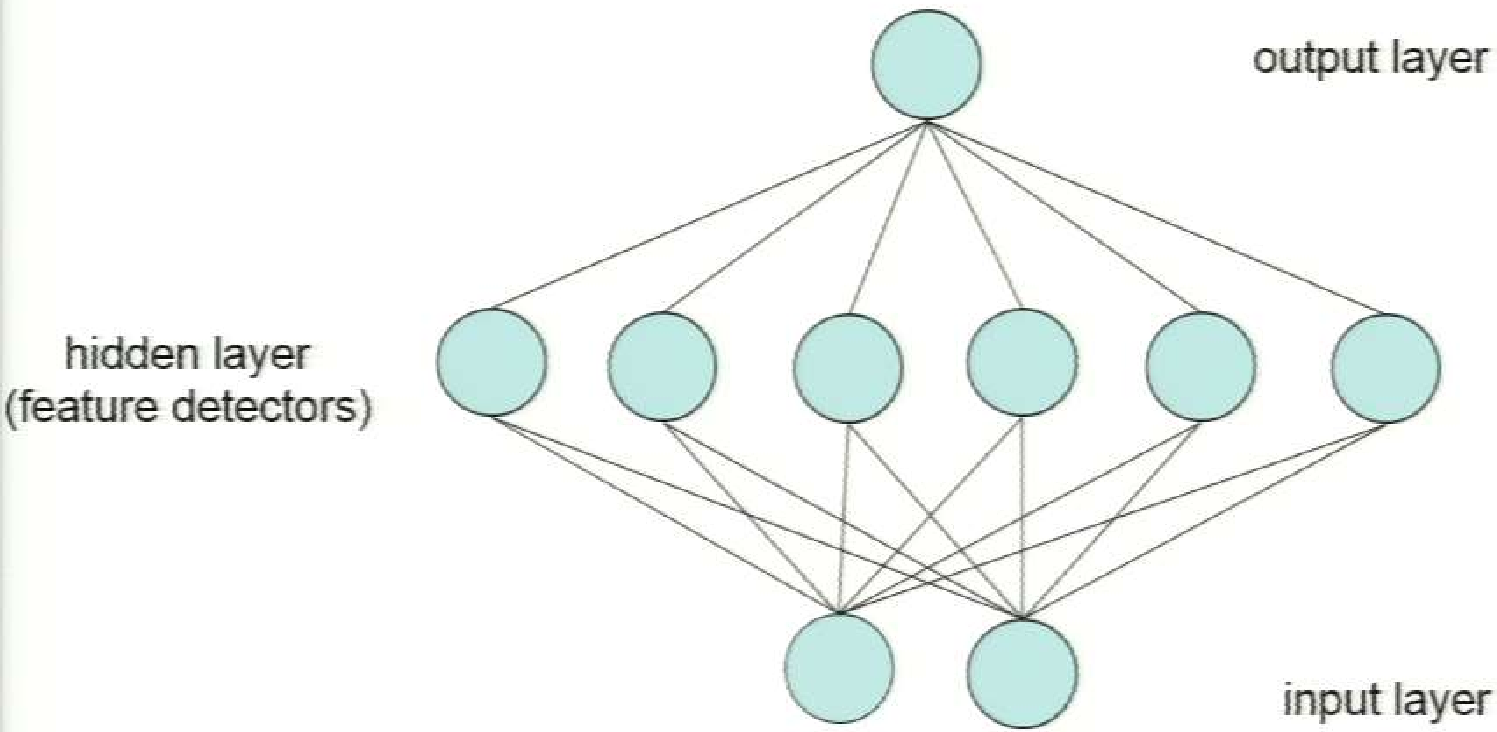
Reviews and Author Rebuttals are now online

http://papers.nips.cc/

Understanding Dropout

P. Baldi and P. Sadowski
University of California, Irvine

Dropout Training



Questions

- Can connections be deleted instead of units?
- Can it be applied to all the layers?
- Can it be used with other values of p?
- What is the optimal p?
- What kind of averaging is dropout implementing?
- What kind of regularization is associated with dropout?
- What are its generalization properties
- Why is it convergent?

Dropout: Linear Networks

- Dropout on units

$$S_i^h = \sum_{l<h} \sum_j w_{ij}^{hl} \delta_j^l S_j^l \quad \text{with} \quad S_j^0 = I_j$$

- Dropout on connections

$$S_i^h = \sum_{l<h} \sum_j \delta_{ij}^{hl} w_{ij}^{hl} S_j^l \quad \text{with} \quad S_j^0 = I_j$$

Dropout: Linear Networks

$$S_i^h = \sum_{l<h} \sum_j w_{ij}^{hl} \delta_j^l S_j^l \quad \text{with} \quad S_j^0 = I_j$$

$$E(S_i^h) = \sum_{l<h} \sum_j w_{ij}^{hl} p_j^l E(S_j^l) \quad \text{for} \quad h > 0$$

- Probabilistic framework allows easy computation of all expectations.
- Probabilistic framework allows easy computation of all variances and covariances:

$$E(S_i^h S_{i'}^{h'}) = E\left[\sum_{l<h} \sum_j w_{ij}^{hl} \delta_j^l S_j^l \sum_{l'<h'} \sum_{j'} w_{i'j'}^{h'l'} \delta_{j'}^{l'} S_{j'}^{l'}\right] = \sum_{l<h} \sum_{l'<h'} \sum_j \sum_{j'} w_{ij}^{hl} w_{i'j'}^{h'l'} E(\delta_j^l \delta_{j'}^{l'}) E(S_j^l S_{j'}^{l'})$$

- Backpercolation.

Dropout: Non-Linear Networks

- Stochastic Network:

$$O_i^h = \sigma_i^h(S_i^h) = \sigma(\sum_{l<h}\sum_j w_{ij}^{hl}\delta_j^l O_j^l) \quad \text{with} \quad O_j^0 = I_j$$

- Deterministic Network:

$$W_i^h = \sigma_i^h(U_i^h) = \sigma(\sum_{l<h}\sum_j w_{ij}^{hl}p_j^l W_j^l) \quad \text{with} \quad W_j^0 = I_j$$

Is the deterministic network computing the ensemble average?

Different Averages

- Real numbers: $0 < O_1, \ldots, O_m < 1$
- Complements: $0 < 1 - O_1, \ldots, 1 - O_m < 1$
- Distribution: $P_1, \ldots, P_m$ with $\sum P_i = 1$

$$E = \sum P_i O_i \quad \text{and} \quad E' = 1 - E = \sum P_i (1 - O_i)$$

$$G = \prod O_i^{P_i} \quad \text{and} \quad G' = \prod (1 - O_i)^{P_i}$$

$$NWGM = \frac{G}{G + G'}$$

Dropout: Non-Linear Networks

$$O = \sigma(S) = \frac{1}{1 + ce^{-\lambda S}}$$

$$NWGM(O(\mathcal{N})) = \frac{\prod_{\mathcal{N}} \sigma(S(\mathcal{N}))^{P(\mathcal{N})}}{\prod_{\mathcal{N}} \sigma(S(\mathcal{N}))^{P(\mathcal{N})} + \prod_{\mathcal{N}} (1 - \sigma(S(\mathcal{N})))^{P(\mathcal{N})}}$$

$$NWGM(O(\mathcal{N})) = \frac{1}{1 + \prod_{\mathcal{N}} (\frac{1-\sigma(S(\mathcal{N}))}{\sigma(S(\mathcal{N})})^{P(\mathcal{N})}} = \frac{1}{1 + ce^{-\lambda \sum_{\mathcal{N}} P(\mathcal{N}) S(\mathcal{N})}} = \sigma(E(S)$$

$$\boxed{NWGM(\sigma(S)) = \sigma(E(S))}$$

Functional Class

Dropout seems to rely on the fundamental property of the logistic sigmoidal function $NWGM(\sigma) = \sigma(E)$. Thus it is natural to wonder what is the class of functions f satisfying this property. *Here we show that the class of functions f defined on the real line with range in $[0, 1]$ and satisfying*

$$\frac{G}{G+G'}(f) = f(E) \tag{59}$$

for any set of points and any distribution, consists exactly of the union of all constant functions $f(x) = K$ with $0 \leq K \leq 1$ and all logistic functions $f(x) = 1/(1 + ce^{-\lambda x})$. As a reminder, G denotes the geometric mean and G' denotes the geometric mean of the complements. Note also that all the constant functions with $f(x) = K$ with $0 \leq K \leq 1$ can also be viewed as logistic functions by taking $\lambda = 0$ and $c = (1 - K)/K$ ($K = 0$ is a limiting case corresponding to $c \to \infty$).

$$\frac{f(u)^p f(v)^{1-p}}{f(u)^p f(v)^{1-p} + (1 - f(u))^p (1 - f(v))^{1-p}} = f(pu + (1-p)v)$$

Dropout: Non-Linear Networks

$$O = \sigma(S) = \frac{1}{1 + ce^{-\lambda S}}$$

$$NWGM(O(\mathcal{N})) = \frac{\prod_{\mathcal{N}} \sigma(S(\mathcal{N}))^{P(\mathcal{N})}}{\prod_{\mathcal{N}} \sigma(S(\mathcal{N}))^{P(\mathcal{N})} + \prod_{\mathcal{N}} (1 - \sigma(S(\mathcal{N})))^{P(\mathcal{N})}}$$

$$NWGM(O(\mathcal{N})) = \frac{1}{1 + \prod_{\mathcal{N}} (\frac{1-\sigma(S(\mathcal{N}))}{\sigma(S(\mathcal{N})})^{P(\mathcal{N})}} = \frac{1}{1 + ce^{-\lambda \sum_{\mathcal{N}} P(\mathcal{N}) S(\mathcal{N})}} = \sigma(E(S)$$

$$\boxed{NWGM(\sigma(S)) = \sigma(E(S))}$$

Functional Class

Dropout seems to rely on the fundamental property of the logistic sigmoidal function $NWGM(\sigma) = \sigma(E)$. Thus it is natural to wonder what is the class of functions f satisfying this property. *Here we show that the class of functions f defined on the real line with range in $[0,1]$ and satisfying*

$$\frac{G}{G+G'}(f) = f(E) \tag{59}$$

for any set of points and any distribution, consists exactly of the union of all constant functions $f(x) = K$ with $0 \leq K \leq 1$ and all logistic functions $f(x) = 1/(1+ce^{-\lambda x})$. As a reminder, G denotes the geometric mean and G' denotes the geometric mean of the complements. Note also that all the constant functions with $f(x) = K$ with $0 \leq K \leq 1$ can also be viewed as logistic functions by taking $\lambda = 0$ and $c = (1-K)/K$ ($K = 0$ is a limiting case corresponding to $c \to \infty$).

$$\frac{f(u)^p f(v)^{1-p}}{f(u)^p f(v)^{1-p} + (1-f(u))^p (1-f(v))^{1-p}} = f(pu + (1-p)v)$$

Dropout Recursion

$$O_i^h = \sigma_i^h(S_i^h) = \sigma(\sum_{l<h}\sum_j w_{ij}^{hl}\delta_j^l O_j^l) \quad \text{with} \quad O_j^0 = I_j$$

$$\boxed{\begin{gathered} E(O_i^h) \approx NWGM(O_i^h) \\ NWGM(O_i^h) = \sigma_i^h\left[E(S_i^h)\right] \\ E(S_i^h) = \sum_{l<h}\sum_j w_{ij}^{hl} p_j^l E(O_j^l) \end{gathered}}$$

1) How good is the approximation of E by the NWGM?
2) How good is the approximation of E by W, i.e. are there systematic errors and do they accumulate or not?

Known Relationships

$$G \le E \quad \text{and} \quad G' \le E'$$

$$\frac{1}{2\max_i O_i} Var(O) \le E - G \le \frac{1}{2\min_i O_i} Var(O)$$ (Cartwright and Fields)

If the numbers O_i satisfy $0 < O_i \le 0.5$ (consistently low), then

(Ky Fan/ Levinson)

$$\frac{G}{G'} \le \frac{E}{E'} \quad \text{and therefore} \quad G \le \frac{G}{G+G'} \le E$$

New Bounds and Estimates

Approach: Expansion around: 0, 1, 0.5, or E

$$G = \prod_i O_i^{P_i} = \prod_i (0.5 + \epsilon_i)^{P_i} = 0.5 \prod_i (1 + 2\epsilon_i)^{P_i}$$

$$G = \frac{1}{2} \prod_i \sum_{n=0}^{\infty} \binom{P_i}{n} (2\epsilon_i)^n = \frac{1}{2} \prod_i \left[1 + P_i 2\epsilon_i + \frac{P_i(P_i - 1)}{2} (2\epsilon_i)^2 + R_3(\epsilon_i) \right]$$

where $R_3(\epsilon_i)$ is the remainder of order three

$$R_3(\epsilon_i) = \binom{P_i}{3} \frac{(2\epsilon_i)^3}{(1 + u_i)^{3 - P_i}} = o(\epsilon_i^2)$$

$$\boxed{G = \frac{1}{2} + \sum_i P_i \epsilon_i + (\sum_i P_i \epsilon_i)^2 - \sum P_i \epsilon_i^2 + o(\epsilon^2) = \frac{1}{2} + E(\epsilon) - Var(\epsilon) + o(\epsilon^2) = E(O) - Var(O) + R_3(\epsilon)}$$

New Bounds and Estimates

To a second order approximation, we have

$$G \approx E - V \quad \text{and} \quad G' \approx 1 - E - V \quad \text{and} \quad \frac{G}{G+G'} \approx \frac{E-V}{1-2V} \quad \text{and} \quad \frac{G'}{G+G'} \approx \frac{1-E-V}{1-2V}$$

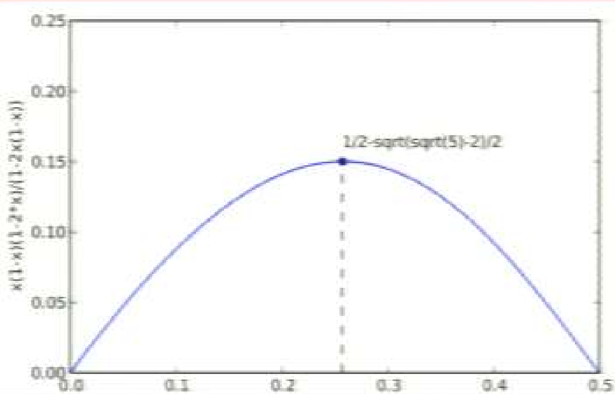
with the differences

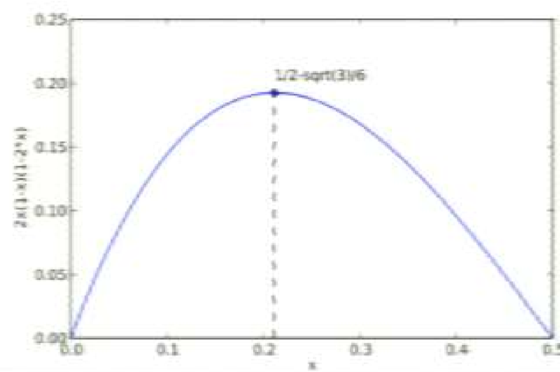
$$|E - \frac{G}{G+G'}| \approx \frac{V(1-2E)}{1-2V} \quad \text{and} \quad |E - \frac{G'}{G+G'}| \approx \frac{V(1-2E)}{1-2V}$$

where V is the variance $V \leq E(1-E)$

$$|E - \frac{G}{G+G'}| \approx \frac{V(1-2E)}{1-2V} \leq \frac{E(1-E)(1-2E)}{1-2E(1-E)} \leq 2E(1-E)(1-2E)$$

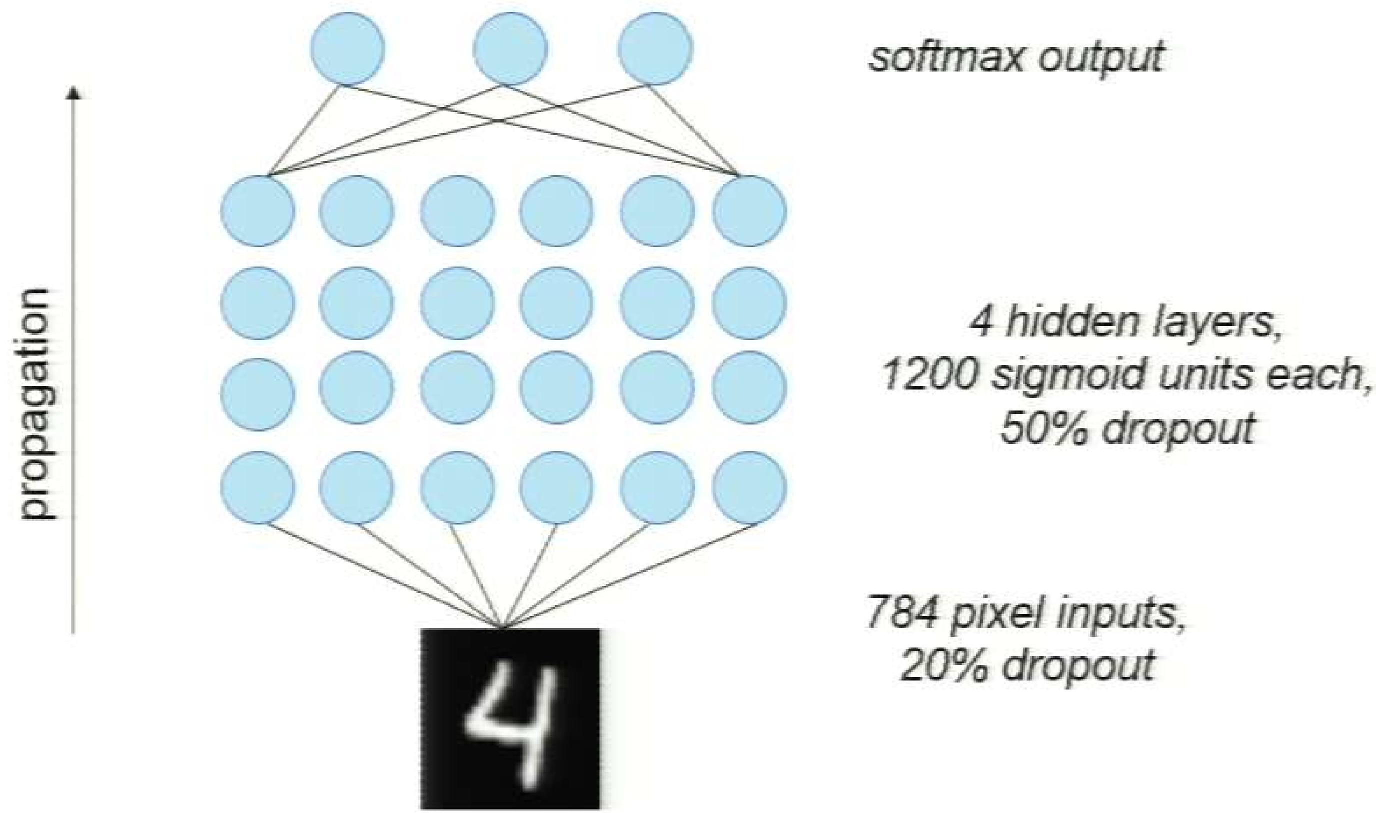
$$|E - \frac{G}{G+G'}| \leq E - G$$





Dropout Simulations

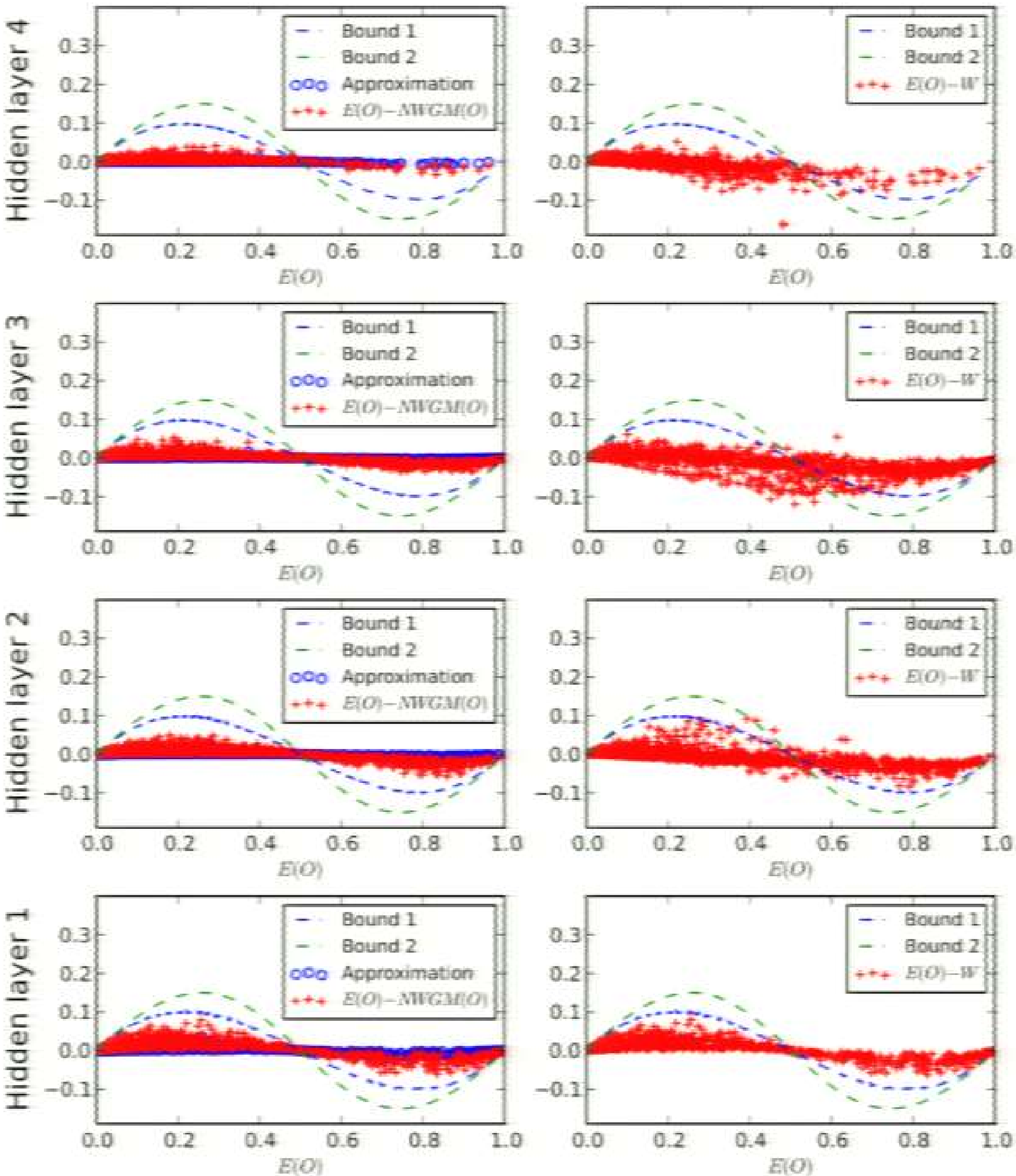
1) Replicated MNIST classifier of Hinton, et. al. 2012
2) Monte Carlo simulations to estimate statistics.



Left: before training

Right: after training

Approximations and bounds are accurate

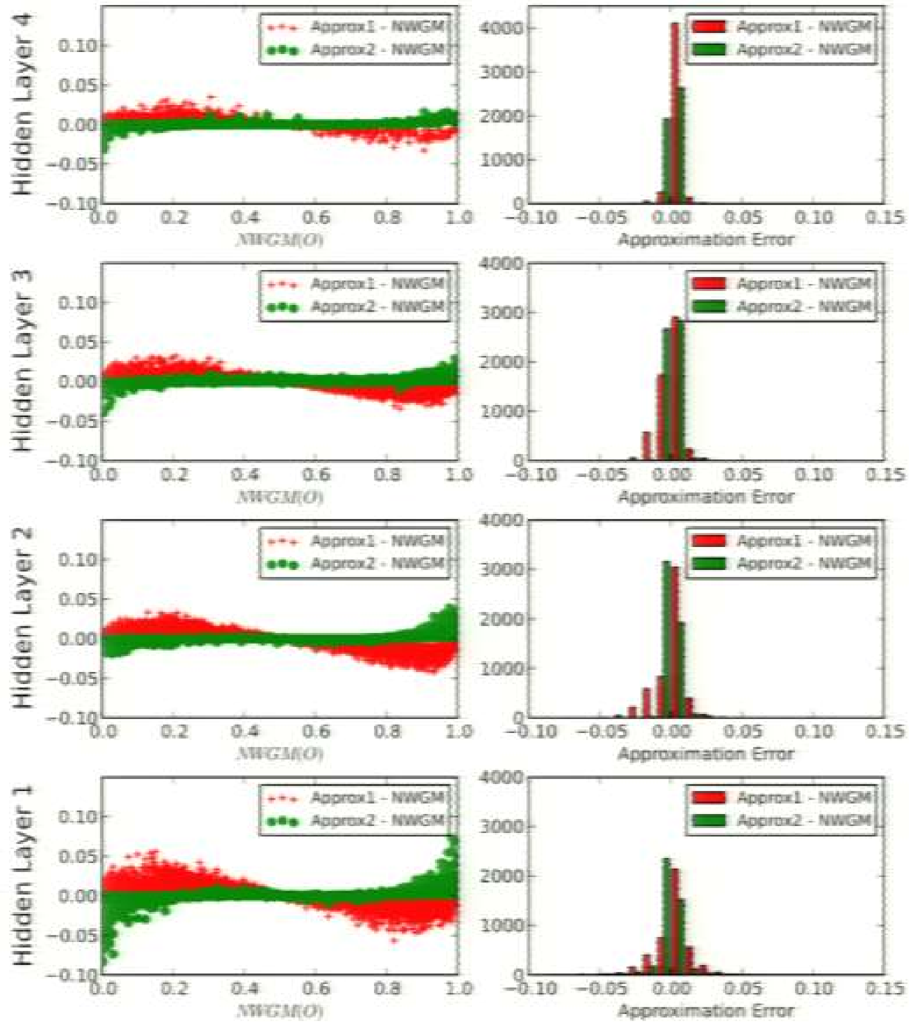


Expansion around 0.5:

$$NWGM = \frac{G}{G+G'} \approx \frac{E-V}{1-2V}$$

Expansion around E:

$$NWGM = \frac{G}{G+G'} \approx \frac{E-\frac{V}{2E}}{1-\frac{1}{2}\frac{V}{E(1-E)}}$$

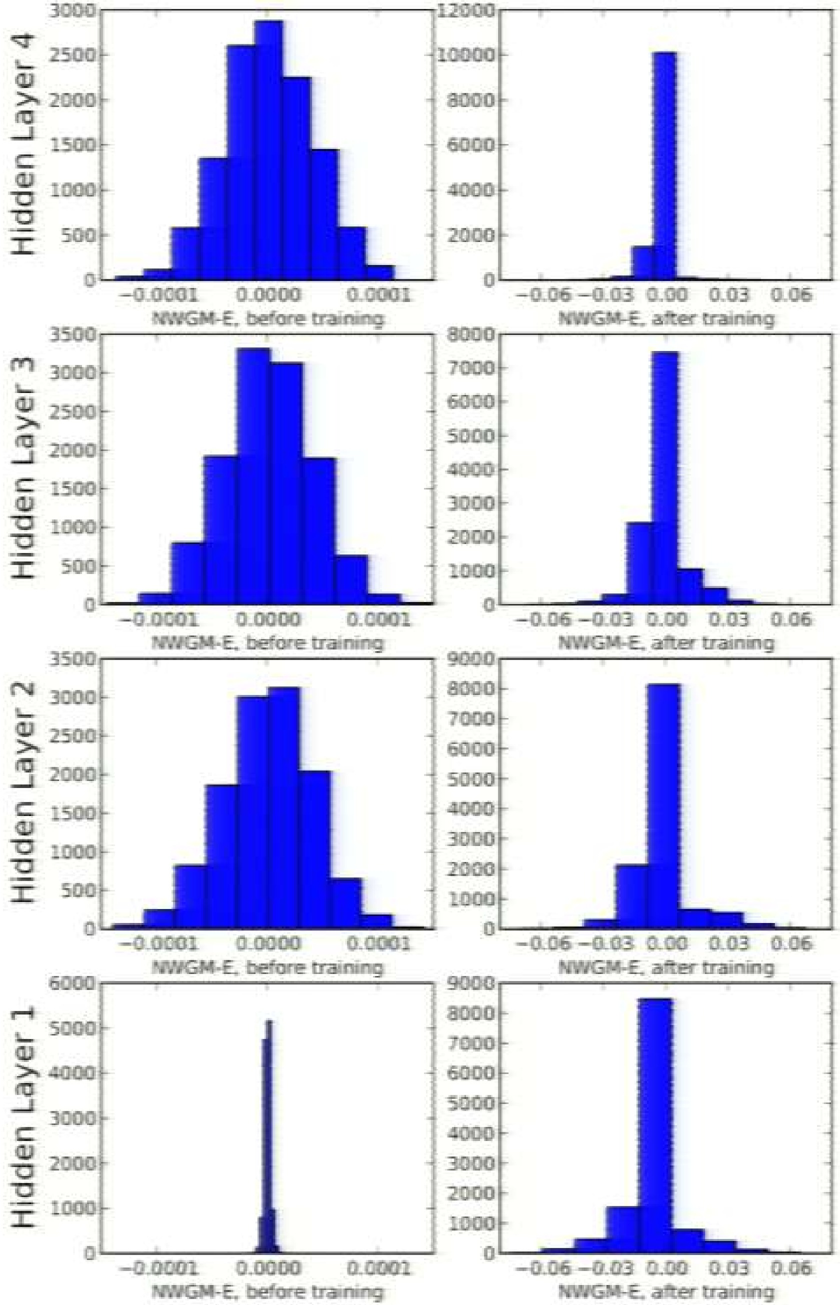


NWGM-E

Left: before training

Right: after training

NWGM is roughly normal around the mean



Errors Do Not Accumulate

- The NWGMs act like approximately Gaussian fluctuations around the true dropout expectations and tend to cancel out.

- [Note: it is always possible to shave off one layer in regression or classification.]

$$Error\left(\frac{\prod_i O_i^{p_i}}{\prod_i O_i^{p_i} + \prod_i (1-O_i)^{p_i}}, t\right) \leq \sum_i p_i Error(O_i, t) \quad \text{or} \quad Error(NWGM) \leq E(Error)$$

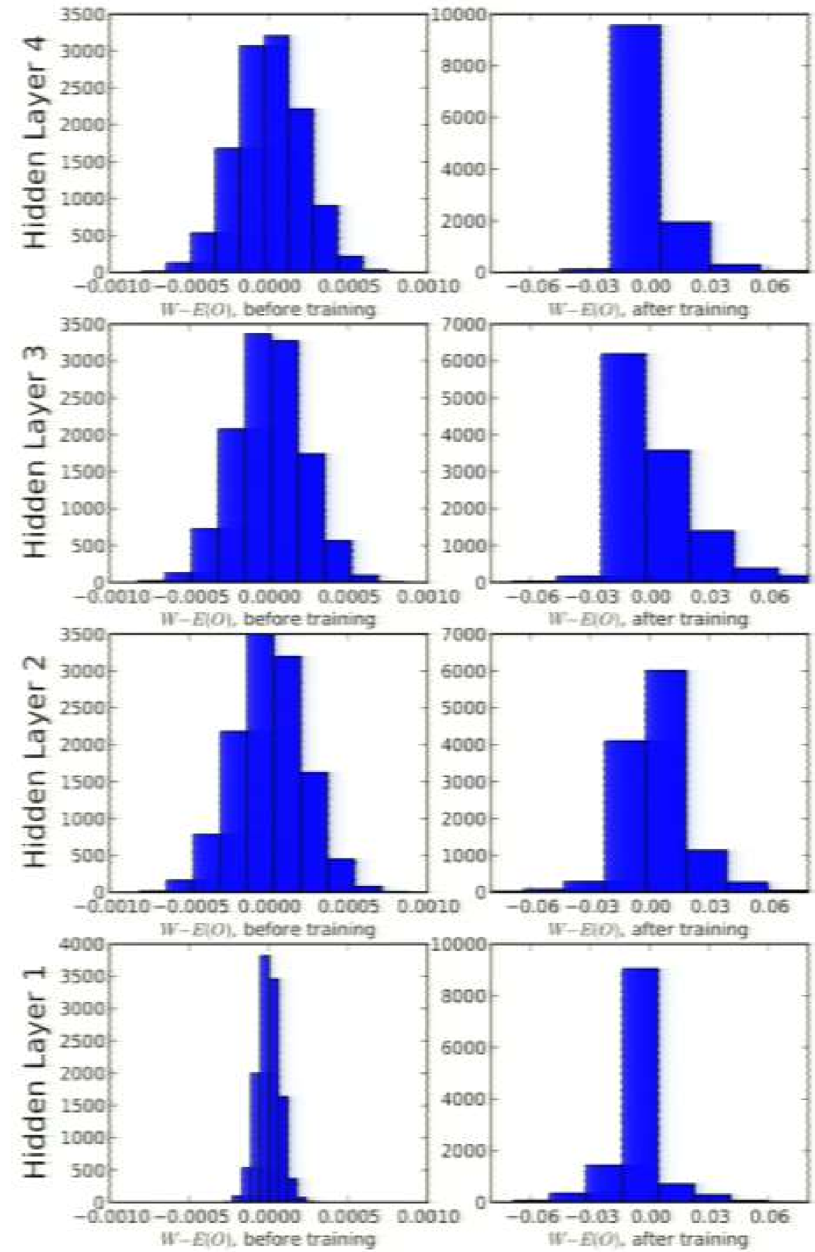
W-E

Left: before training

Right: after training

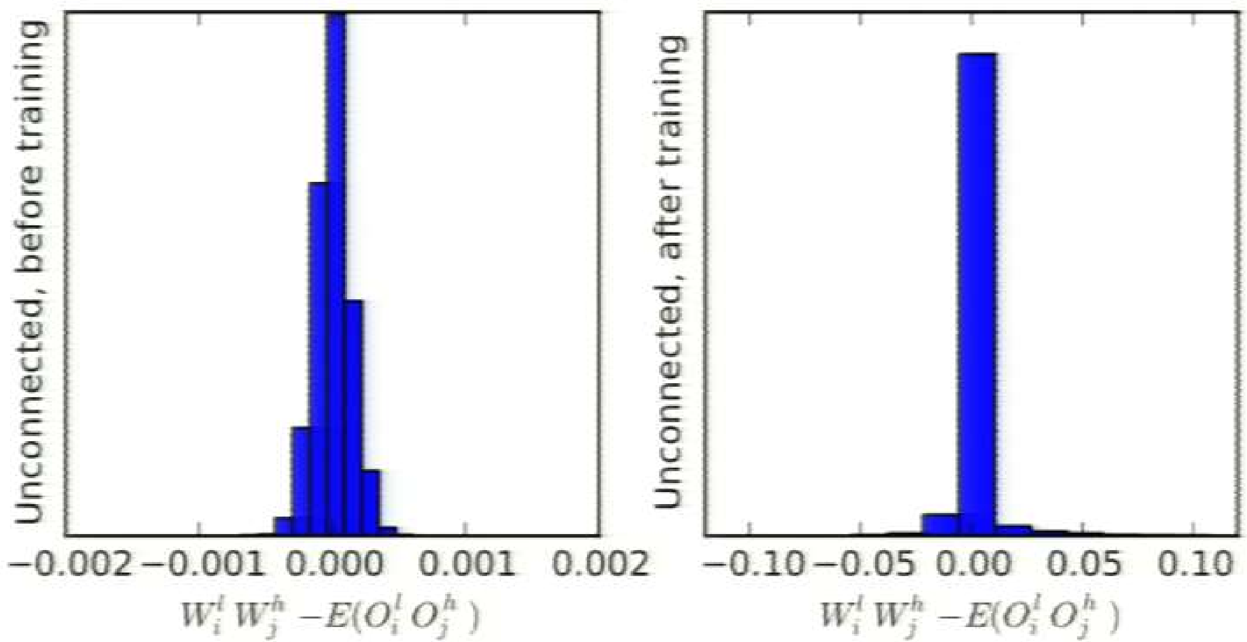
Result:

Approximation error is small (<0.1), even in upper layers.



Higher Order Moments

$$E(O_i^l O_j^h) = E(O_i^l)E(O_j^h) \approx W_i^l W_j^h$$



Dropout Adaptive Regularization

- Linear Case:

$$E_D = \frac{1}{2}(t - O_D)^2 = \frac{1}{2}(t - \sum_{i=1}^{n} \delta_i w_i I_i)^2$$

$$E_{ENS} = \frac{1}{2}(t - O_{ENS})^2 = \frac{1}{2}(t - \sum_{i=1}^{n} p_i w_i I_i)^2$$

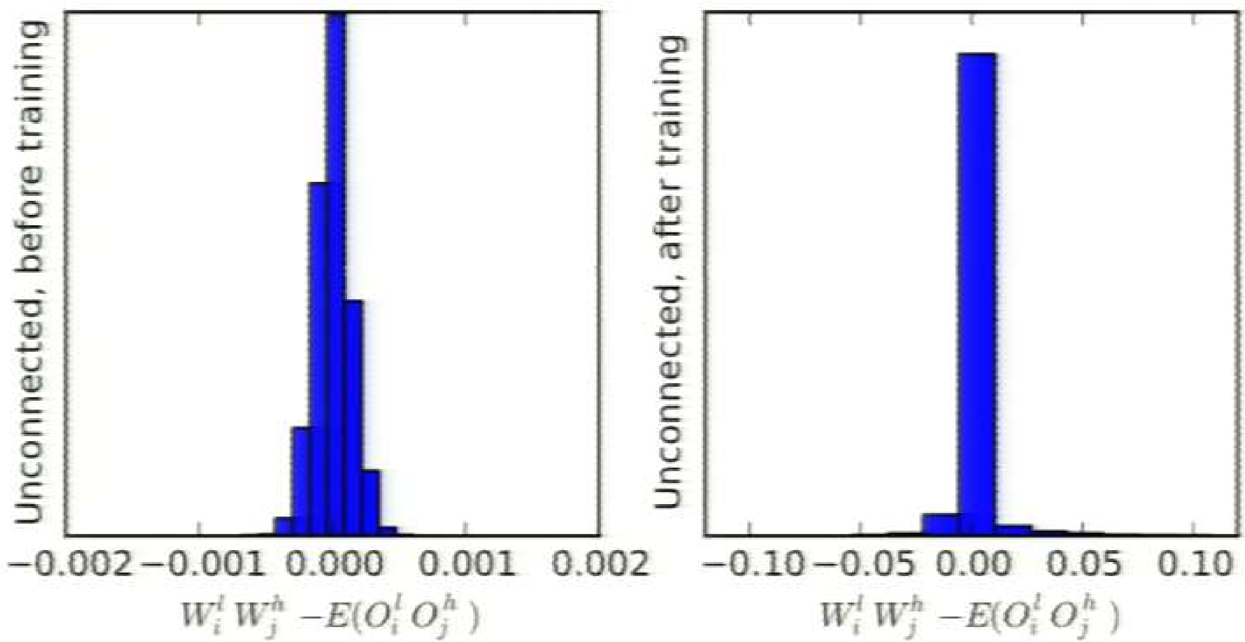
$$\frac{\partial E_D}{\partial w_i} = -(t - O_D)\delta_i I_i = -t\delta_i I_i + w_i \delta_i^2 I_i^2 + \sum_{j \neq i} w_j \delta_i \delta_j I_i I_j$$

$$E\left(\frac{\partial E_D}{\partial w_i}\right) = \frac{\partial E_{ENS}}{\partial w_i} + w_i I_i^2 Var\delta_i = \frac{\partial E_{ENS}}{\partial w_i} + w_i Var(\delta_i I_i)$$

$$\boxed{E = E_{ENS} + \frac{1}{2}\sum_{i=1}^{n} w_i^2 I_i^2 Var\delta_i}$$

Higher Order Moments

$$E(O_i^l O_j^h) = E(O_i^l)E(O_j^h) \approx W_i^l W_j^h$$



Dropout Adaptive Regularization

- Linear Case:

$$E_D = \frac{1}{2}(t - O_D)^2 = \frac{1}{2}(t - \sum_{i=1}^{n} \delta_i w_i I_i)^2$$

$$E_{ENS} = \frac{1}{2}(t - O_{ENS})^2 = \frac{1}{2}(t - \sum_{i=1}^{n} p_i w_i I_i)^2$$

$$\frac{\partial E_D}{\partial w_i} = -(t - O_D)\delta_i I_i = -t\delta_i I_i + w_i \delta_i^2 I_i^2 + \sum_{j \neq i} w_j \delta_i \delta_j I_i I_j$$

$$E\left(\frac{\partial E_D}{\partial w_i}\right) = \frac{\partial E_{ENS}}{\partial w_i} + w_i I_i^2 Var\delta_i = \frac{\partial E_{ENS}}{\partial w_i} + w_i Var(\delta_i I_i)$$

$$\boxed{E = E_{ENS} + \frac{1}{2}\sum_{i=1}^{n} w_i^2 I_i^2 Var\delta_i}$$

Dropout Adaptive Regularization

- Non-Linear Case:

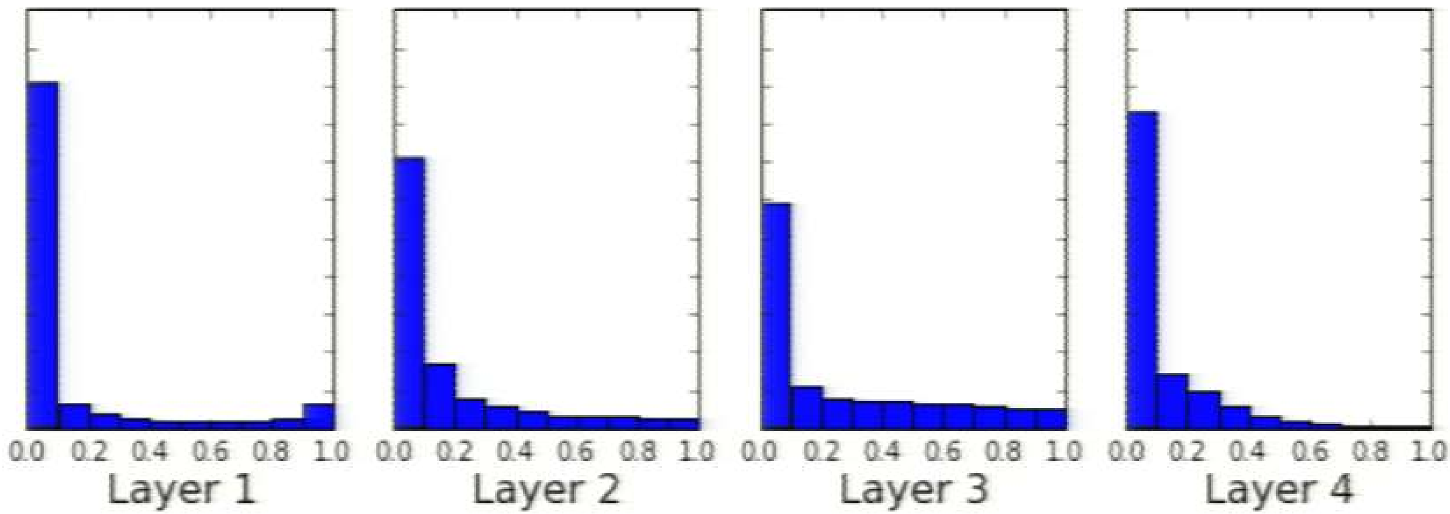
$$\frac{\partial E_D}{\partial w_i} = -\lambda(t - O_D)\delta_i I_i = \lambda\left(t - \sigma(\sum_j w_j \delta_j I_j)\right)\delta_i I_i$$

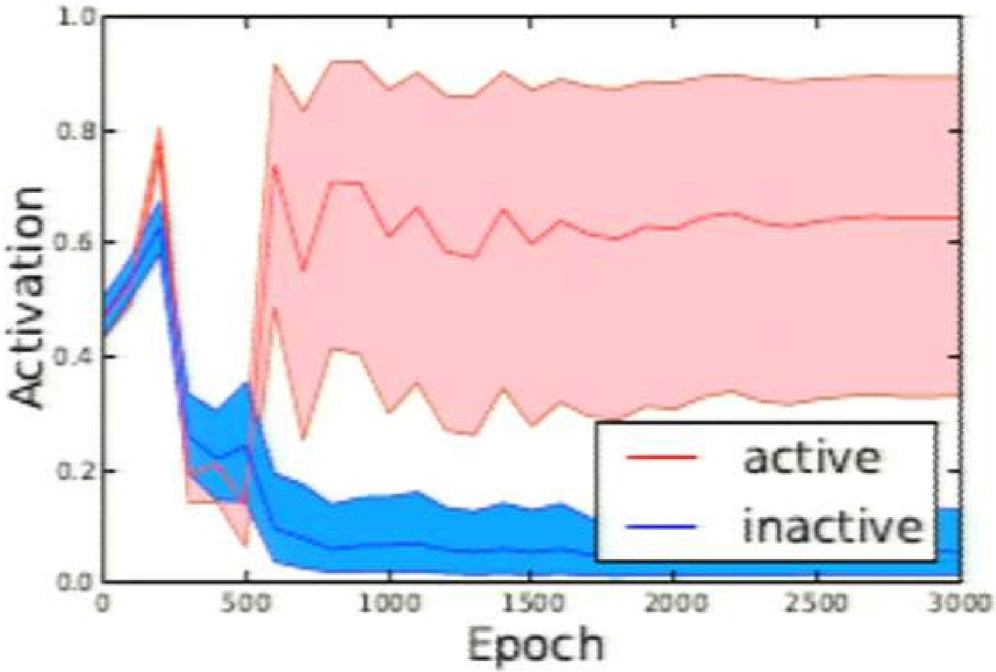
$$E\left(\frac{\partial E_D}{\partial w_i}\right) \approx \frac{\partial E_{ENS}}{\partial w_i} + \lambda\sigma'(S_{ENS}) w_i I_i^2 Var(\delta_i)$$

$$\boxed{E = E_{ENS} + \frac{1}{2}\lambda\sigma'(S_{ENS})\sum_{i=1}^{n} w_i^2 I_i^2 Var(\delta_i)}$$

Simulation Results: Sparsity

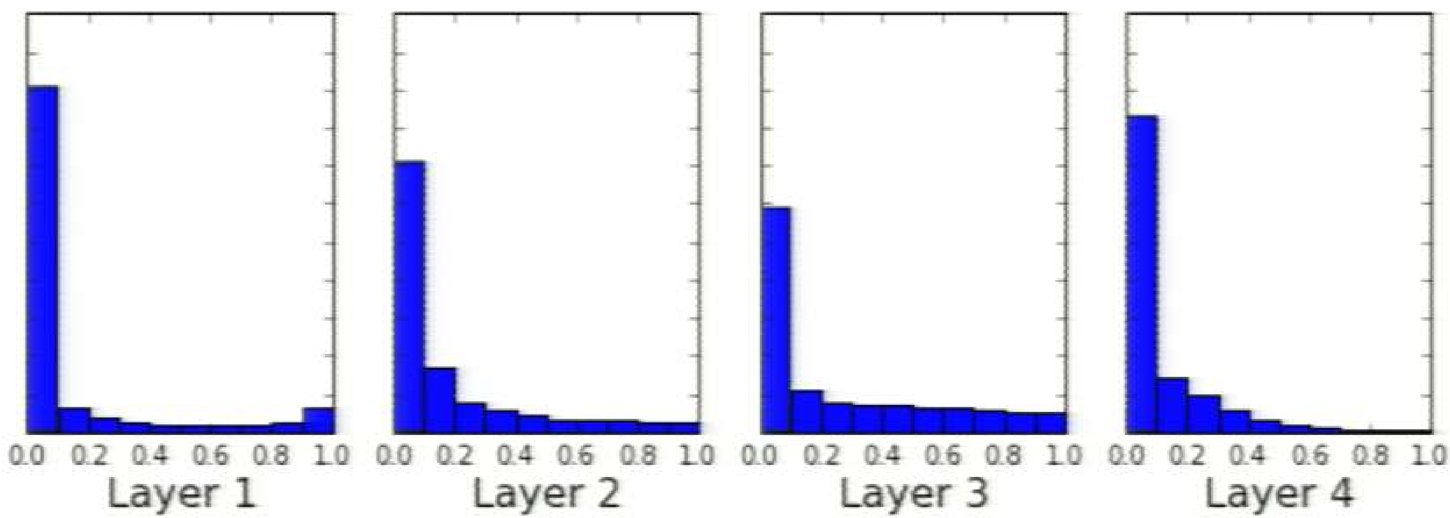
Distribution of neuron activations:

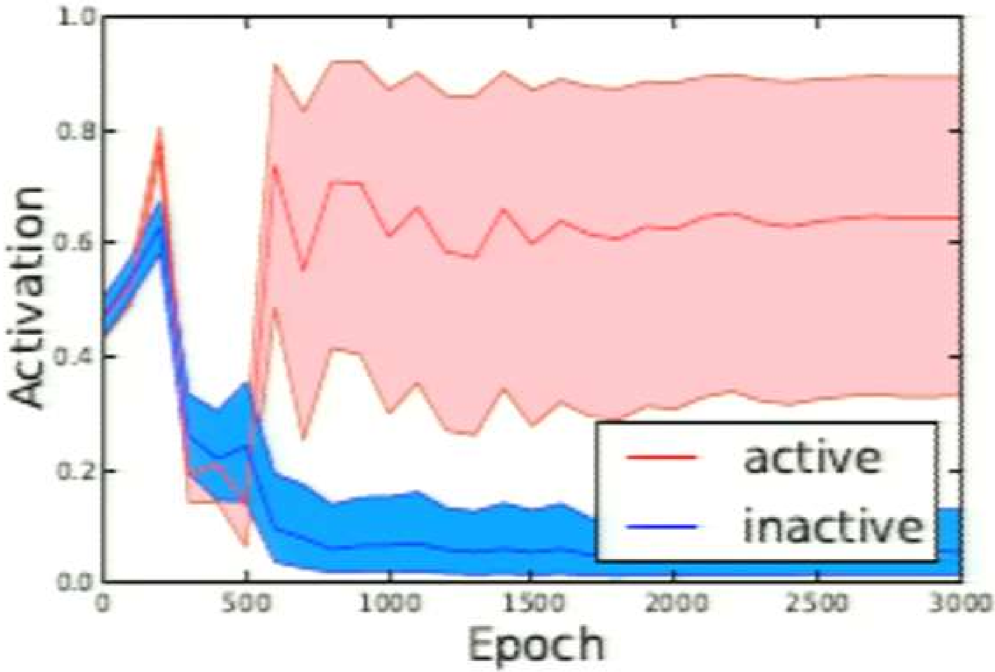



1.0
0.8
0.6
0.4
0.2
0.0
Activation
0
500
1000
1500
2000
2500
3000
Epoch
active
inactive

Simulation Results: Sparsity

Distribution of neuron activations:




1.0
0.8
0.6
0.4
0.2
0.0
Activation
0
500
1000
1500
2000
2500
3000
Epoch
active
inactive

10th Community Wide Experiment on the
Critical Assessment of Techniques for Protein Structure Prediction

RR Analysis

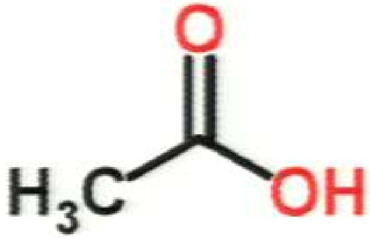
The table summarizes the evaluation of predictions in 'RR' category.
The analysis was performed at per domains basis; only predictions for domains classified as "FM", "TBM/FM", "TBM hard" were considered.
The groups were ranked according to sum of average Z-scores for two measures Acc and Xd.
The per target Z-scores were recalculated from the "cleaned" distributions, where the outlier predictions (below mean - 2 std dev) were eliminated.

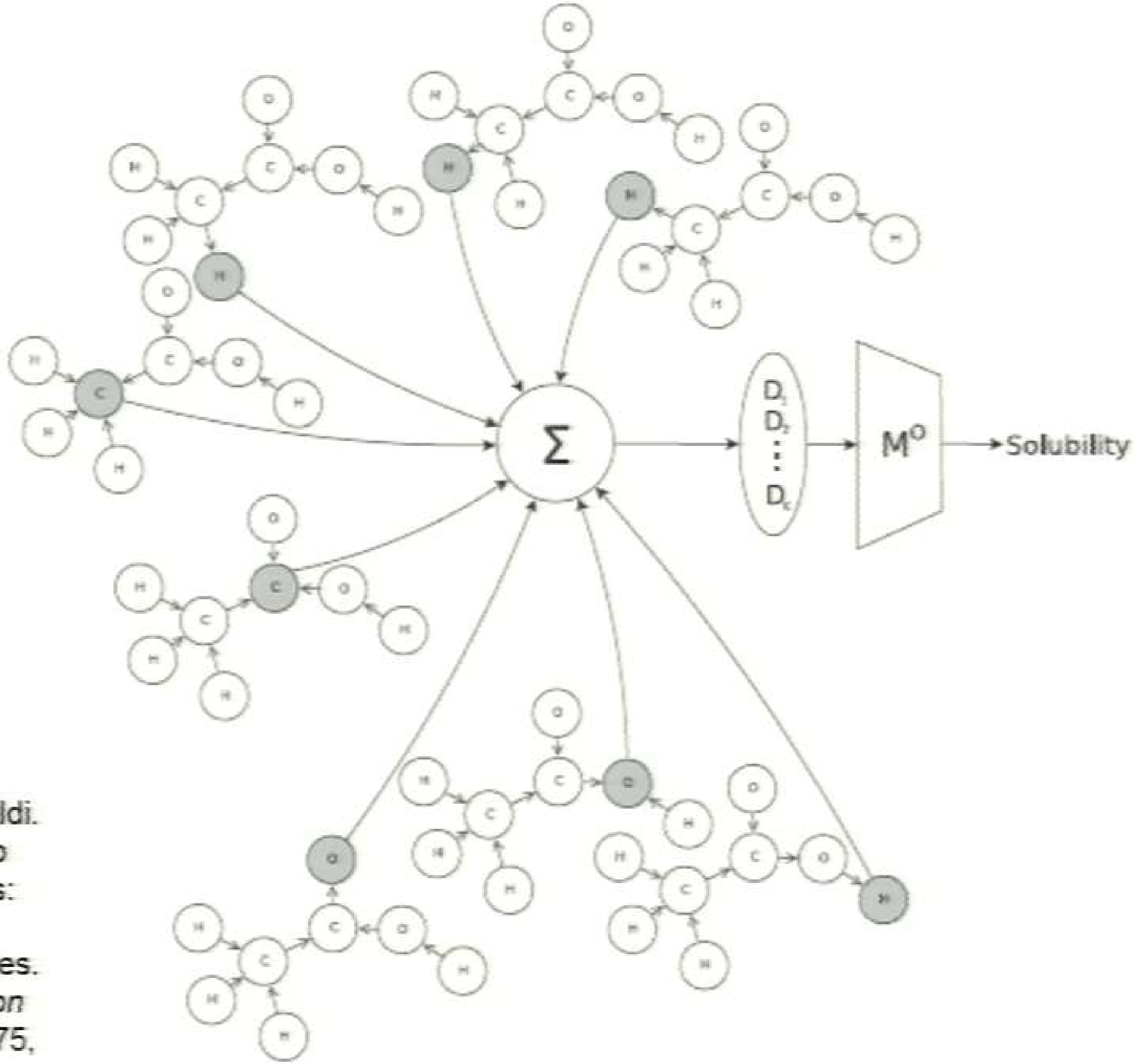
- **Domain classification:**
  - FM
  - TBM/FM
  - TBM hard (max gdt_ts < 50)
  -
- **Contact Range:** long
- **List Size:** L/5

P. Di Lena, K. Nagata, and P. Baldi. Deep Architectures for Protein Contact Map Prediction. *Bioinformatics*, 28, 2449-2457, (2012).

#	GR#	GR Name	Count domains	AVG Acc	AVG Zscore Acc	AVG Xd	AVG Zscore Xd	Zscore Acc + Zscore Xd
1.	222 s	MULTICOM-CONSTRUCT	14	19.41	0.58	12.08	0.77	1.35
2.	305 s	IGBteam	15	19.22	0.72	10.19	0.58	1.30
3.	424 s	MULTICOM-NOVEL	14	20.39	0.50	10.32	0.72	1.22
4.	125 s	MULTICOM-REFINE	14	21.35	0.51	10.29	0.70	1.21
5.	413 s	ZHOU-SPARKS-X	12	12.26	0.62	8.26	0.59	1.21
6.	113 s	SAM-T08-server	11	16.13	0.72	9.44	0.47	1.19
7.	358 s	RaptorX-Roll	8	12.07	0.58	8.23	0.55	1.13
8.	314 s	ProC_S4	14	17.91	0.59	9.76	0.47	1.05
9.	087 s	Distill_roll	15	13.97	0.60	8.57	0.36	0.96
10.	489	MULTICOM	14	12.96	0.43	8.19	0.40	0.83
11.	184 s	ICOS	14	17.03	0.40	9.72	0.39	0.78
12.	396 s	ProC_S5	14	16.51	0.36	9.10	0.36	0.72
13.	381 s	SAM-T06-[illegible]	10	10.98	0.37	7.94	0.31	0.68

Deep Learning





A. Lusci, G. Pollastri, and P. Baldi. Deep Architectures and Deep Learning in Chemoinformatics: the Prediction of Aqueous Solubility for Drug-Like Molecules. *Journal of Chemical Information and Modeling*, 53, 7, 1563–1575, (2013).

Questions

- Can connections be deleted instead of units?
- Can it be applied to all the layers?
- Can it be used with other values of p?
- What is the optimal p?
- What kind of averaging is dropout implementing?
- What kind of regularization is associated with dropout?
- What are its generalization properties
- Why is it convergent?

NIPS Thanks Its Sponsors

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

amazon.com

Microsoft Research

Google

facebook

SKYTREE
THE MACHINE LEARNING COMPANY

TWO SIGMA

United Technologies Research Center

D E Shaw & Co

xerox

IBM Research

YAHOO! LABS

TOYOTA

DRW TRADING GROUP

millionshort

criteo

PDT PARTNERS

Springer
Machine Learning Journal

Disney Research

Annealing between distributions by averaging moments

Roger Grosse

Chris Maddison

Ruslan Salakhutdinov

Motivation

- Would you trust an algorithm that hasn't been validated?
- This is the position we're in for density modeling!

Motivation

- Would you trust an algorithm that hasn't been validated?
- This is the position we're in for density modeling!
- Markov random fields

$$p(\mathbf{x}) = \frac{f(\mathbf{x})}{\mathcal{Z}}$$
$$\mathcal{Z} = \sum_{\mathbf{x}} f(\mathbf{x})$$

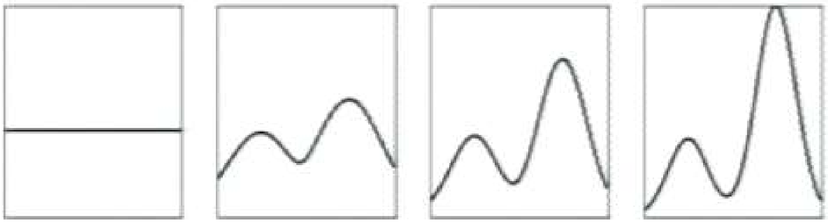
- **Evaluating the likelihood requires estimating the intractable** $\mathcal{Z}$

Motivation

- Many algorithms sample from sequences of distributions
  - bridge from tractable p_{init} to intractable p_{tgt}
  - e.g. **annealed importance sampling**, path sampling, thermodynamic integration, tempered transitions, parallel tempering, nested sampling
- Typical choice: geometric averages

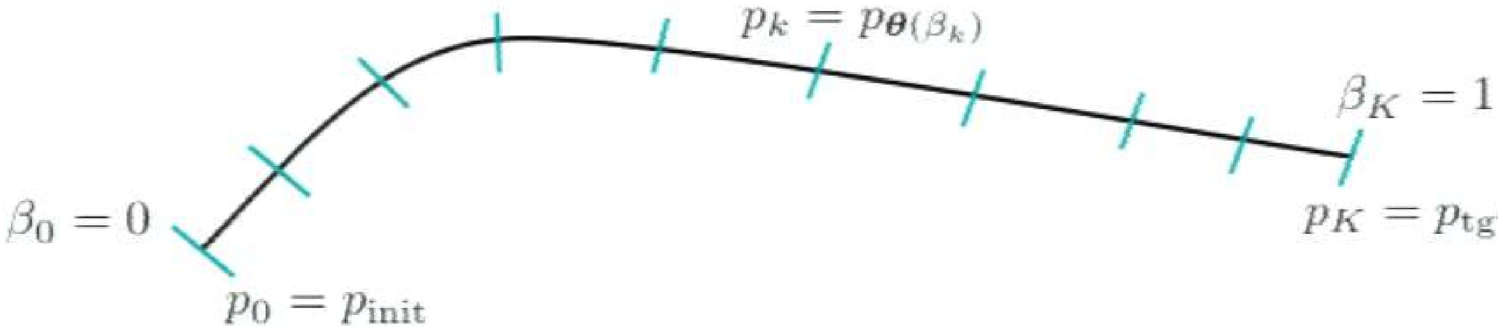
$$p_\beta(\mathbf{x}) \propto p_{\mathrm{init}}(\mathbf{x})^{1-\beta} p_{\mathrm{tgt}}(\mathbf{x})^{\beta}$$

- "Annealing" effect



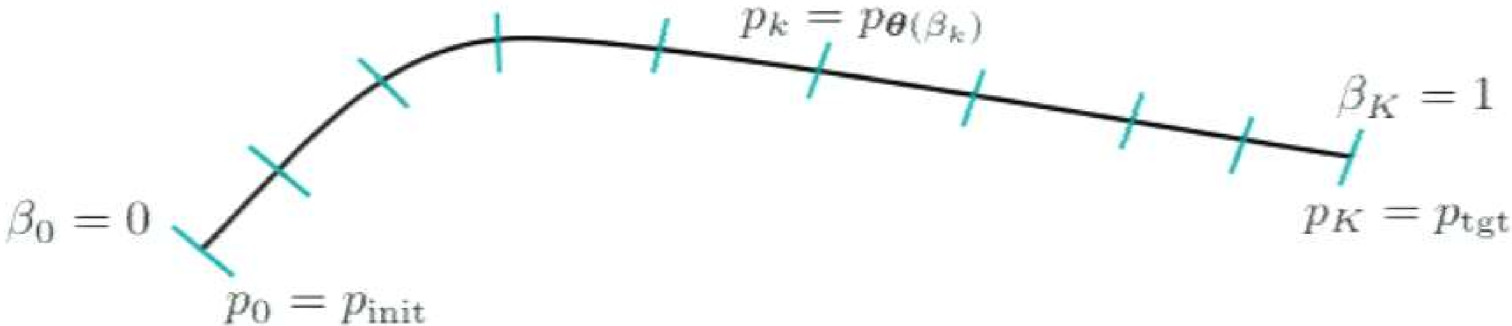
Annealing paths

- Let $\mathcal{P}$ be a family of distributions parameterized by $\boldsymbol{\theta}$
- Annealing path $\gamma : [0, 1] \to \mathcal{P}$

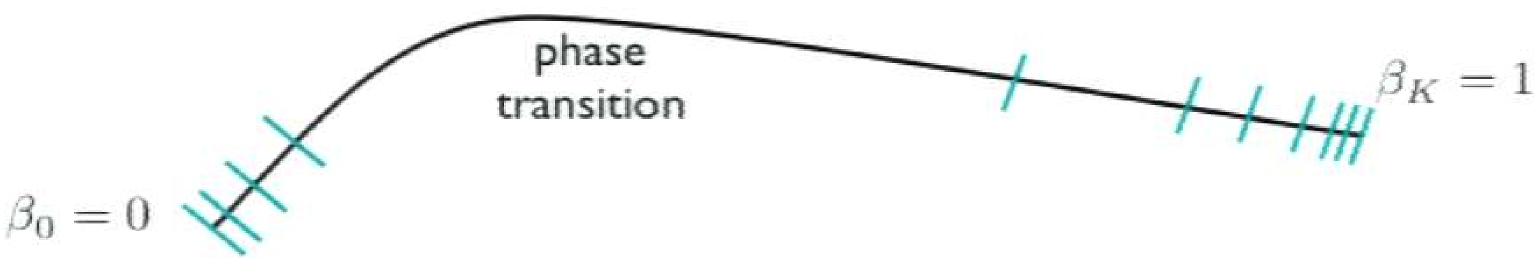


Annealing paths

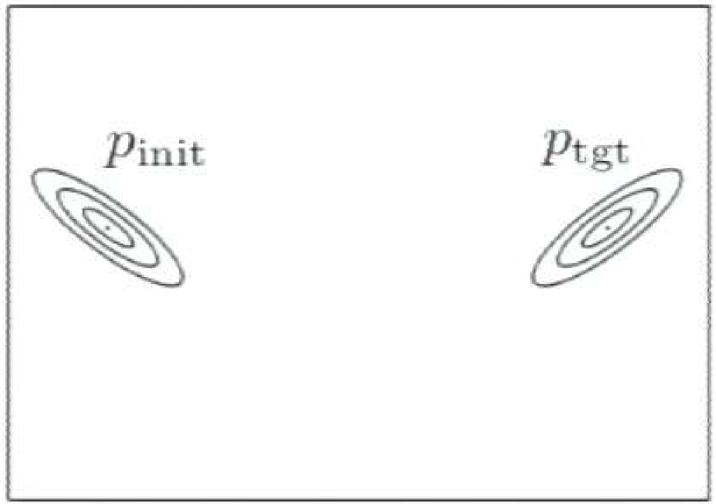
- Let $\mathcal{P}$ be a family of distributions parameterized by $\boldsymbol{\theta}$
- Annealing path $\gamma : [0, 1] \to \mathcal{P}$



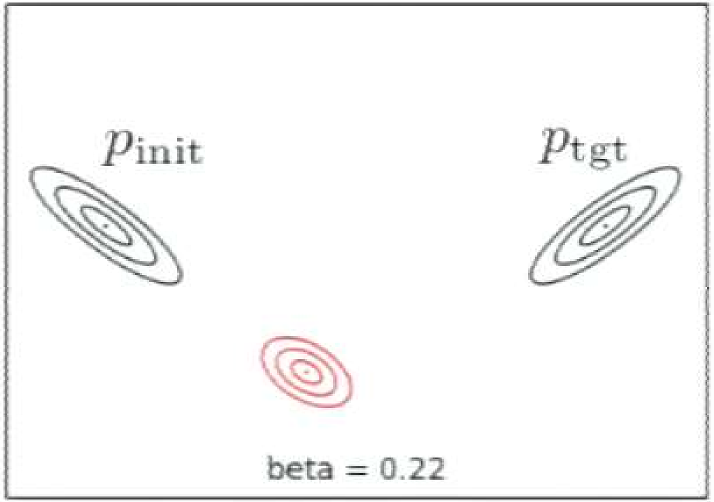
A more honest cartoon:



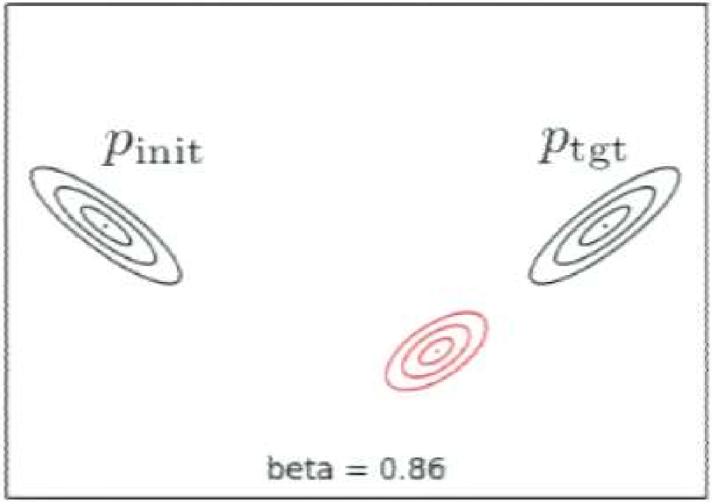
Geometric averages can be counterintuitive



Geometric averages can be counterintuitive

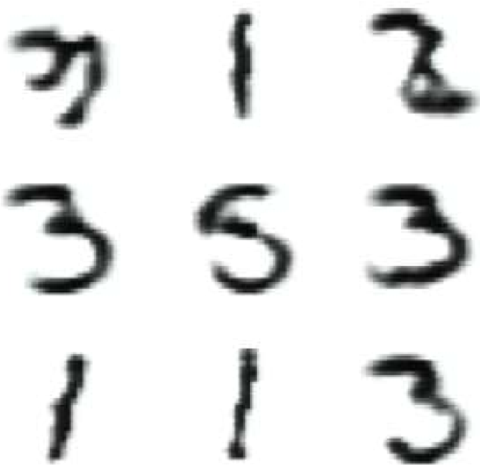


Geometric averages can be counterintuitive

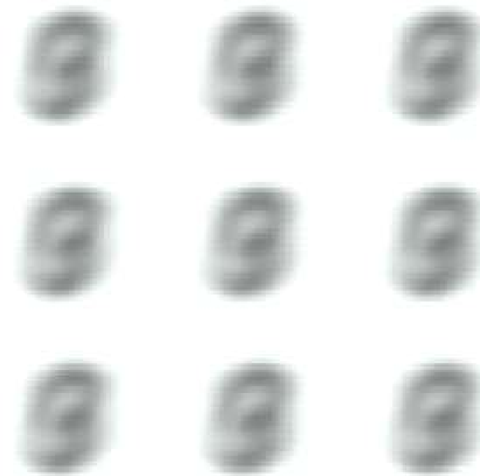


Geometric averages can be counterintuitive

RBM trained to MNIST



samples from
target distribution

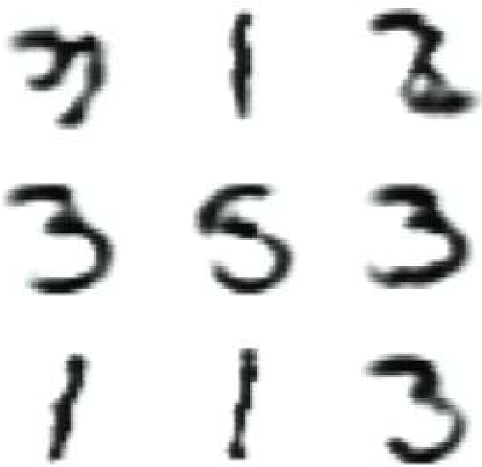


beta = 0.00

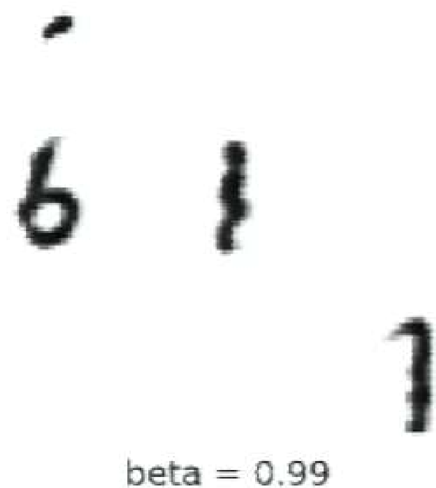
geometric
averages

Geometric averages can be counterintuitive

RBM trained to MNIST



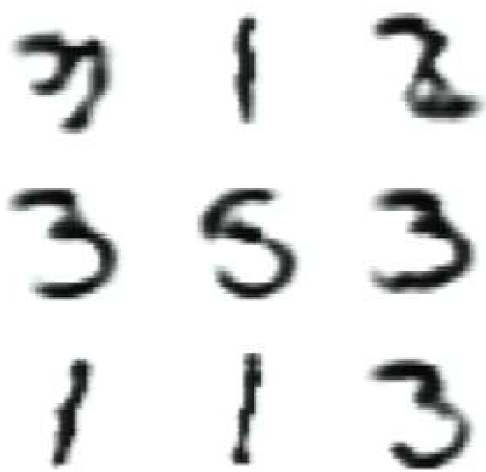
samples from
target distribution



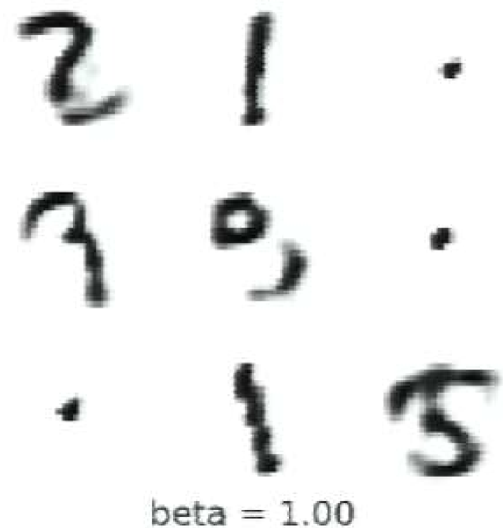
geometric
averages

Geometric averages can be counterintuitive

RBM trained to MNIST



samples from
target distribution



geometric
averages

Annealed importance sampling

Given:

unnormalized distributions $f_0, \ldots, f_K$

MCMC transition operators $\mathcal{T}_0, \ldots, \mathcal{T}_K$

$f_0 = f_{\text{init}}$ easy to sample from, compute partition function of

$\mathbf{x} \sim f_{\text{init}}$

$w = \mathcal{Z}_{\text{init}}$

For $i = 0, \ldots, K-1$

$w := w \frac{f_{i+1}(\mathbf{x})}{f_i(\mathbf{x})}$

$\mathbf{x} :\sim \mathcal{T}_{i+1}(\cdot \,|\, \mathbf{x})$

Then, $\mathbb{E}[w] = \mathcal{Z}_{\text{tgt}}$

$$\hat{\mathcal{Z}}_{\text{tgt}} = \frac{1}{S} \sum_{s=1}^{S} w^{(s)}$$

Be careful estimating partition functions!

- AIS gives an unbiased estimate of $\mathcal{Z}_{\mathrm{tgt}}$

$$\mathbb{E}[\hat{\mathcal{Z}}_{\mathrm{tgt}}] = \mathcal{Z}_{\mathrm{tgt}}$$

- But it gives a biased estimate of $\log \mathcal{Z}_{\mathrm{tgt}}$

$$\mathbb{E}[\log \hat{\mathcal{Z}}_{\mathrm{tgt}}] \leq \log \mathcal{Z}_{\mathrm{tgt}}$$

Be careful estimating partition functions!

- AIS gives an unbiased estimate of $\mathcal{Z}_{\mathrm{tgt}}$

$$\mathbb{E}[\hat{\mathcal{Z}}_{\mathrm{tgt}}] = \mathcal{Z}_{\mathrm{tgt}}$$

- But it gives a biased estimate of $\log \mathcal{Z}_{\mathrm{tgt}}$

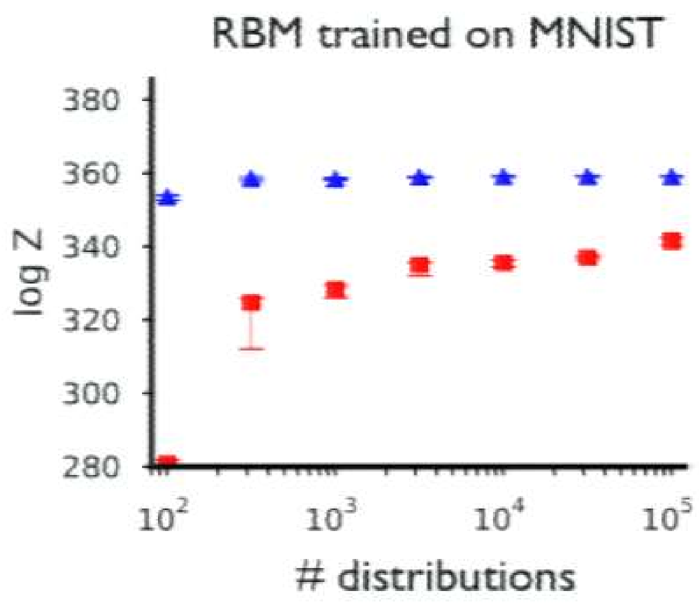
$$\mathbb{E}[\log \hat{\mathcal{Z}}_{\mathrm{tgt}}] \leq \log \mathcal{Z}_{\mathrm{tgt}}$$

- Do you have a good model or a bad partition function estimator?

$$\text{overestimate} \longrightarrow p_{\mathrm{tgt}}(\mathbf{x}) = \frac{f_{\mathrm{tgt}}(\mathbf{x})}{\mathcal{Z}_{\mathrm{tgt}} \longleftarrow \text{underestimate}}$$

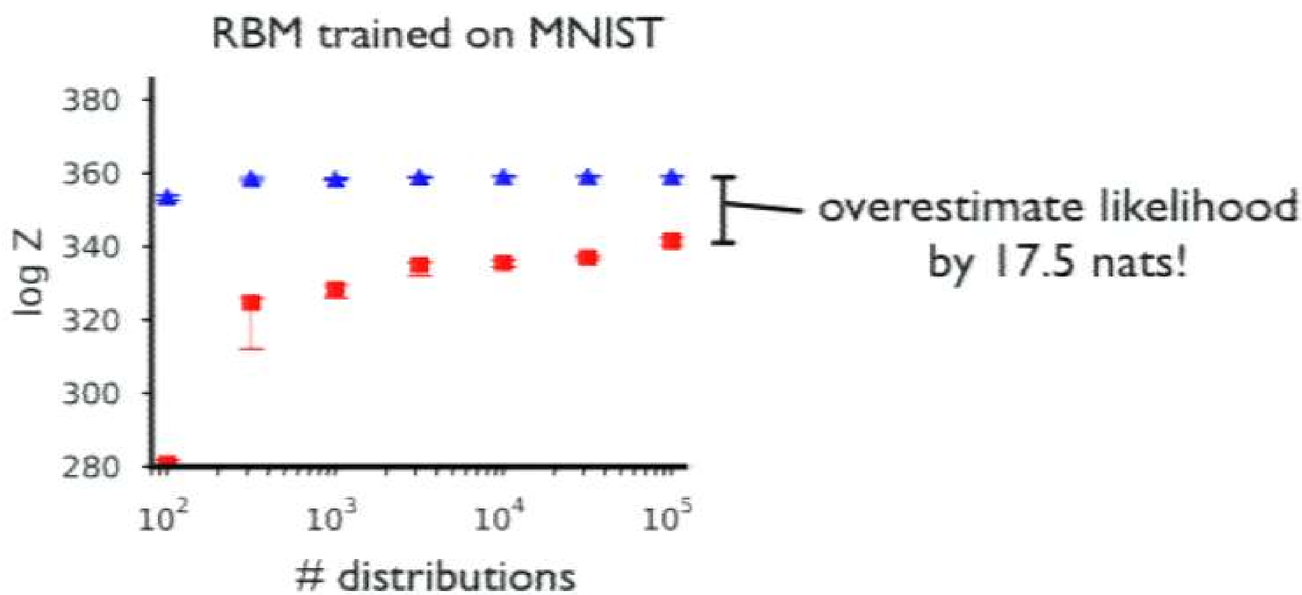
Be careful estimating partition functions!

- Is this a problem in practice?



Be careful estimating partition functions!

- Is this a problem in practice?



Moment averaging

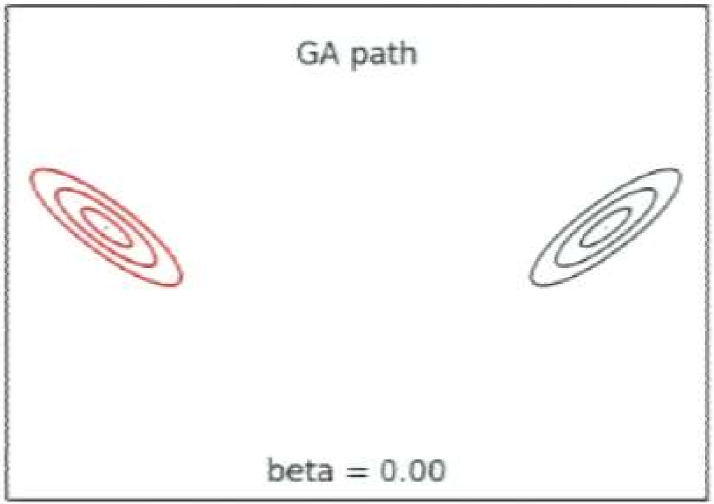
- Exponential families

$$p(\mathbf{x}) = \frac{1}{\mathcal{Z}(\boldsymbol{\eta})} h(\mathbf{x}) \exp\left(\boldsymbol{\eta}^T \mathbf{g}(\mathbf{x})\right)$$

- Two equivalent representations
  - natural parameters $\boldsymbol{\eta}$
  - moments $\mathbf{s} = \mathbb{E}[\mathbf{g}(\mathbf{x})]$
- Averaging the natural parameters = geometric averages

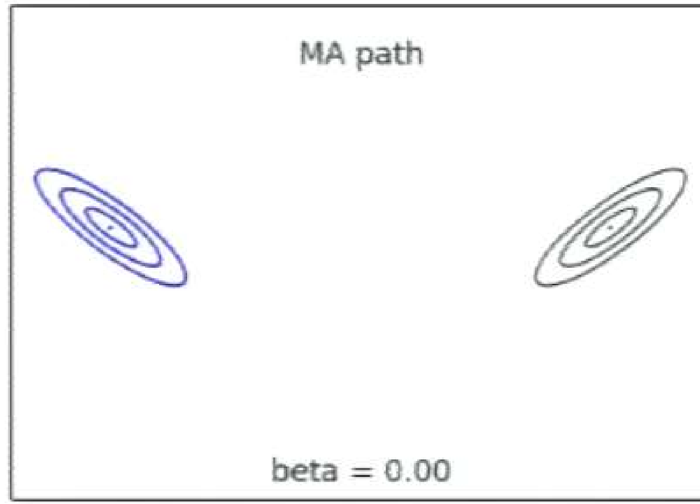
$$\boldsymbol{\eta}(\beta) = (1 - \beta)\boldsymbol{\eta}_{\text{init}} + \beta \boldsymbol{\eta}_{\text{tgt}}$$

Moment averaging



$$\boldsymbol{\eta}(\beta) = (1-\beta)\boldsymbol{\eta}_{\mathrm{init}} + \beta\boldsymbol{\eta}_{\mathrm{tgt}}$$

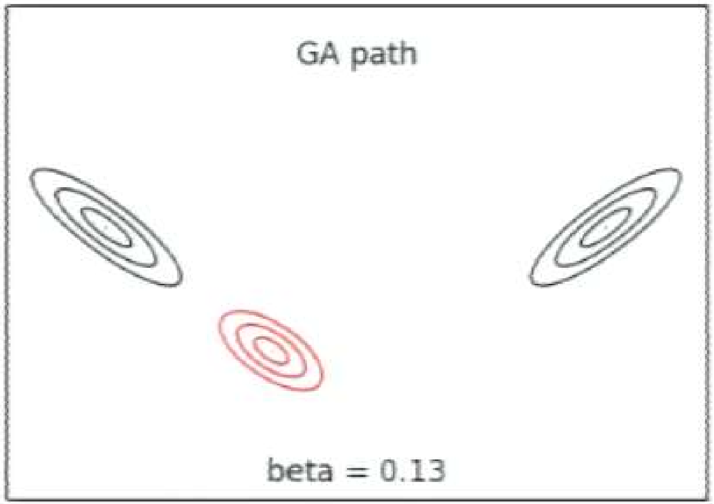
Geometric averages

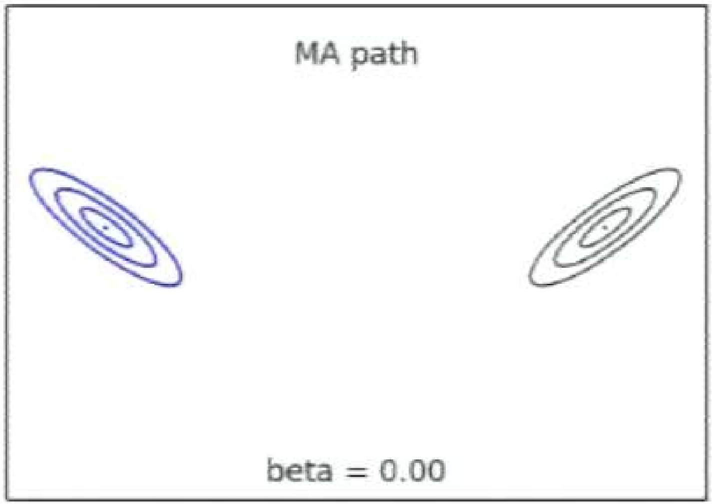


$$\mathbf{s}(\beta) = (1-\beta)\mathbf{s}_{\mathrm{init}} + \beta\mathbf{s}_{\mathrm{tgt}}$$

Moment averages

Moment averaging





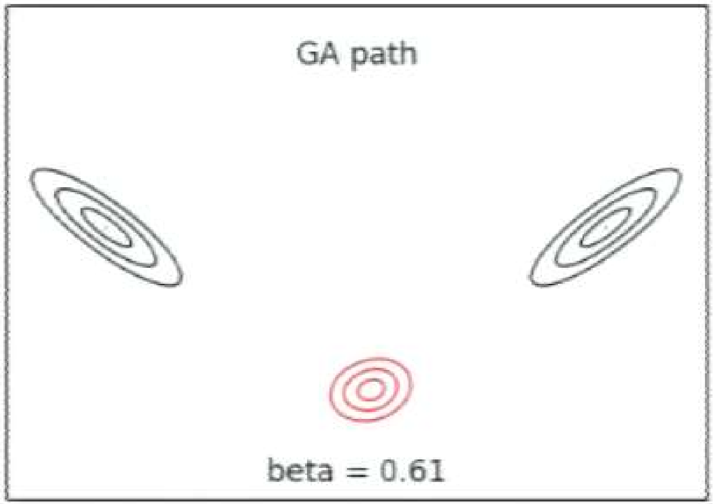
$$\boldsymbol{\eta}(\beta) = (1-\beta)\boldsymbol{\eta}_{\text{init}} + \beta\boldsymbol{\eta}_{\text{tgt}}$$

Geometric averages

$$\mathbf{s}(\beta) = (1-\beta)\mathbf{s}_{\text{init}} + \beta\mathbf{s}_{\text{tgt}}$$

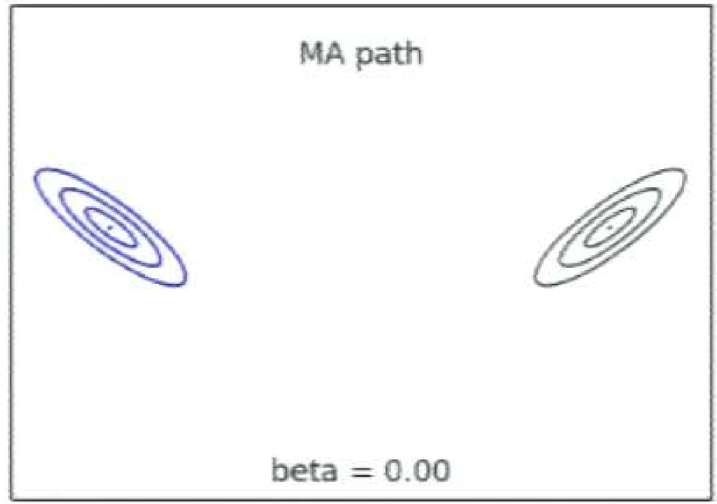
Moment averages

Moment averaging



$$\boldsymbol{\eta}(\beta) = (1-\beta)\boldsymbol{\eta}_{\text{init}} + \beta\boldsymbol{\eta}_{\text{tgt}}$$

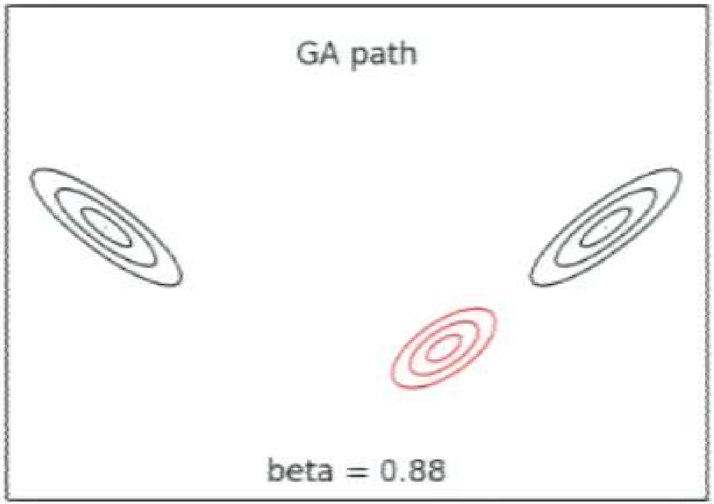
Geometric averages



$$\mathbf{s}(\beta) = (1-\beta)\mathbf{s}_{\text{init}} + \beta\mathbf{s}_{\text{tgt}}$$

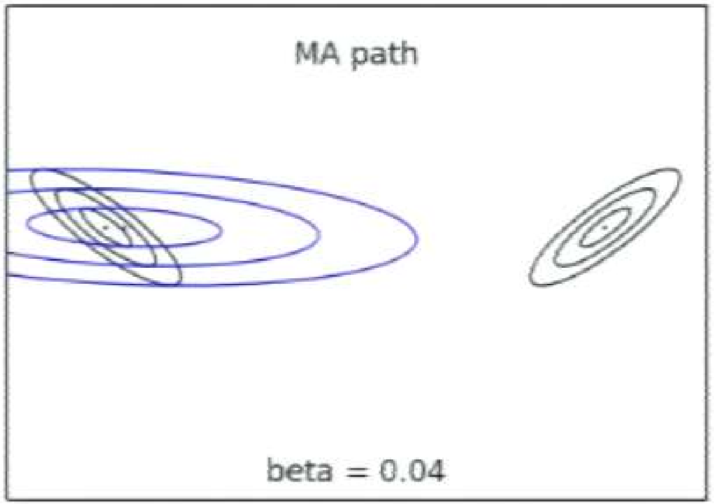
Moment averages

Moment averaging



$$\boldsymbol{\eta}(\beta) = (1-\beta)\boldsymbol{\eta}_{\rm init} + \beta\boldsymbol{\eta}_{\rm tgt}$$

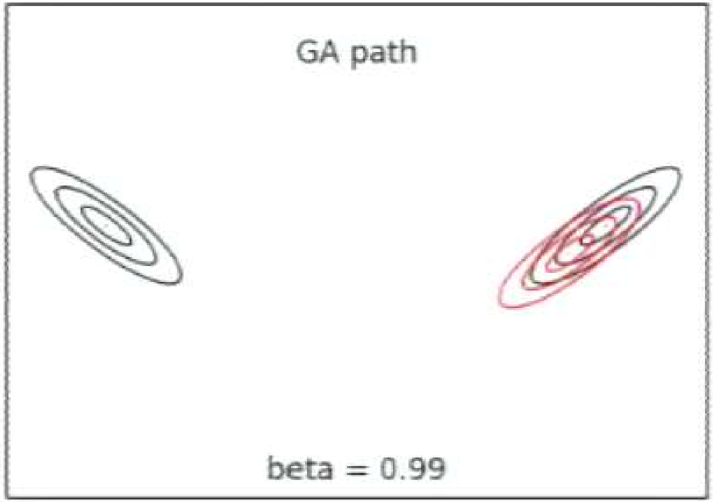
Geometric averages

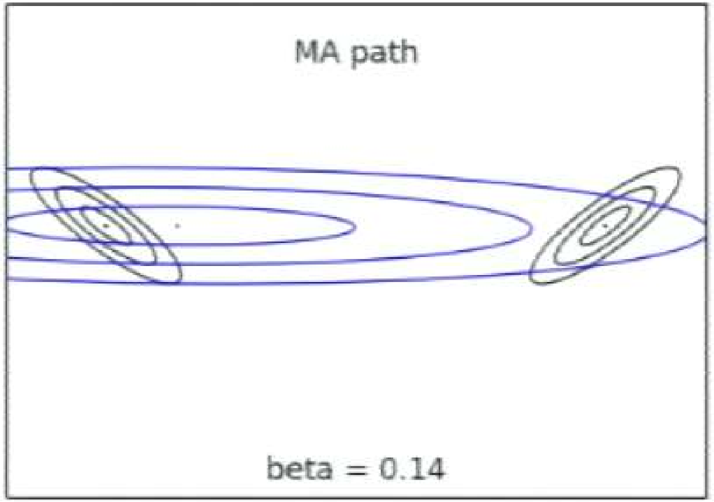


$$\mathbf{s}(\beta) = (1-\beta)\mathbf{s}_{\rm init} + \beta\mathbf{s}_{\rm tgt}$$

Moment averages

Moment averaging





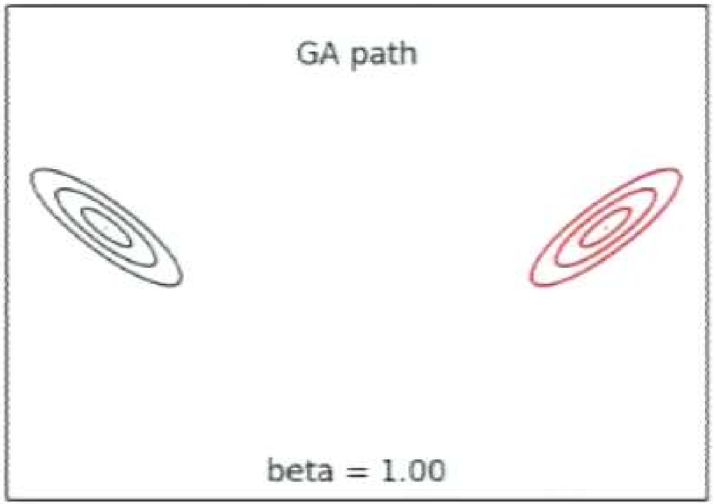
$$\boldsymbol{\eta}(\beta) = (1-\beta)\boldsymbol{\eta}_{\text{init}} + \beta\boldsymbol{\eta}_{\text{tgt}}$$

Geometric averages

$$\mathbf{s}(\beta) = (1-\beta)\mathbf{s}_{\text{init}} + \beta\mathbf{s}_{\text{tgt}}$$

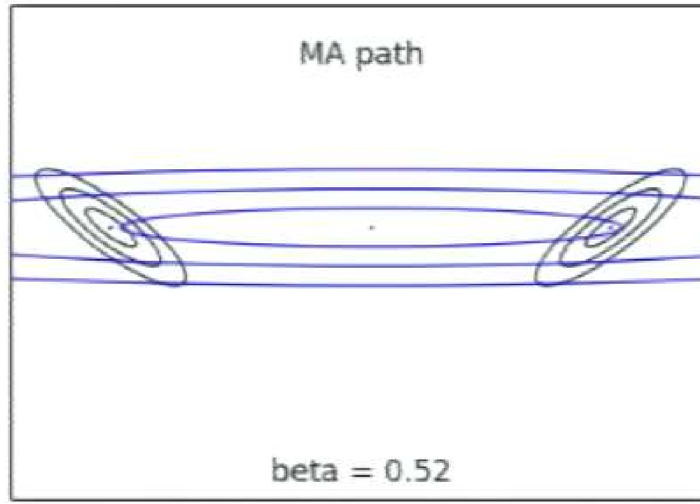
Moment averages

Moment averaging



$$\boldsymbol{\eta}(\beta) = (1-\beta)\boldsymbol{\eta}_{\text{init}} + \beta\boldsymbol{\eta}_{\text{tgt}}$$

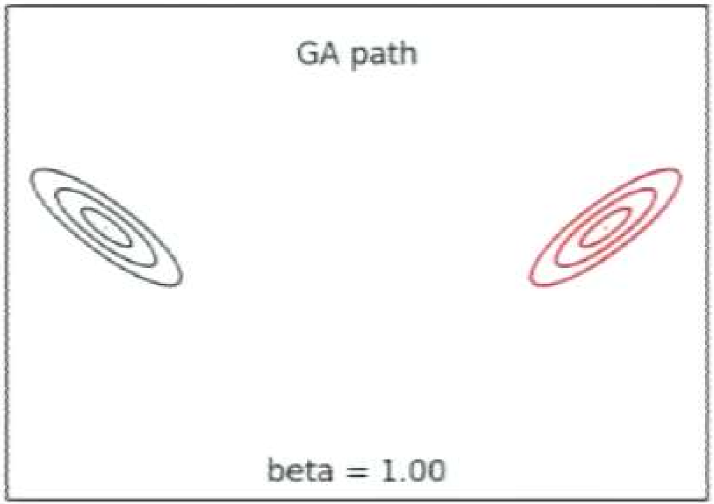
Geometric averages

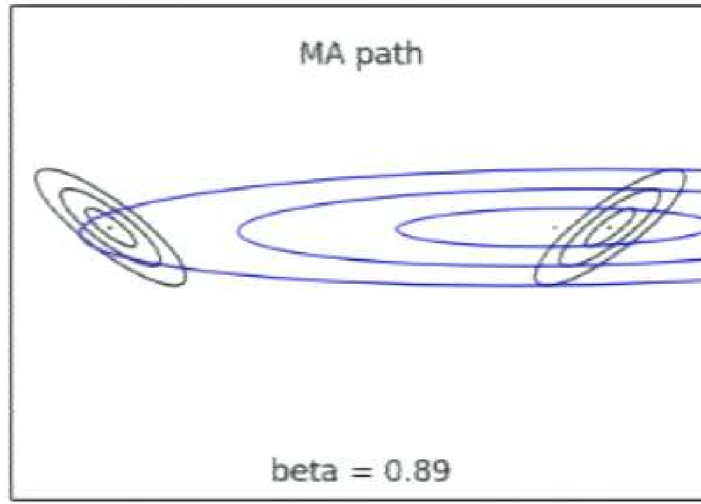


$$\mathbf{s}(\beta) = (1-\beta)\mathbf{s}_{\text{init}} + \beta\mathbf{s}_{\text{tgt}}$$

Moment averages

Moment averaging





$$\boldsymbol{\eta}(\beta) = (1-\beta)\boldsymbol{\eta}_{\text{init}} + \beta\boldsymbol{\eta}_{\text{tgt}}$$

Geometric averages

$$\mathbf{s}(\beta) = (1-\beta)\mathbf{s}_{\text{init}} + \beta\mathbf{s}_{\text{tgt}}$$

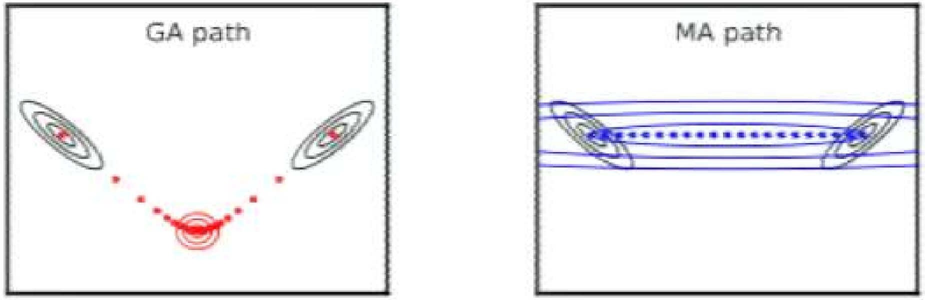
Moment averages

Moment averaging

- Variational interpretation of GA and MA paths:

$$p_\beta^{(\mathrm{GA})} = \arg\min_q \, (1-\beta)\mathrm{D_{KL}}(q \| p_{\mathrm{init}}) + \beta \mathrm{D_{KL}}(q \| p_{\mathrm{tgt}})$$

$$p_\beta^{(\mathrm{MA})} = \arg\min_q \, (1-\beta)\mathrm{D_{KL}}(p_{\mathrm{init}} \| q) + \beta \mathrm{D_{KL}}(p_{\mathrm{tgt}} \| q)$$



- MA tries to cover *all* modes of target distribution

Analyzing AIS paths

- Can analyze bias analytically
  - assume perfect transitions (MCMC operator returns an exact sample)

$$\mathbb{E}\left[\log w\right] = \log \mathcal{Z}_{\mathrm{init}} + \sum_{i=0}^{K-1} \mathbb{E}_{p_i}\left[\log f_{i+1}(\mathbf{x}) - \log f_i(\mathbf{x})\right]$$

$$= \log \mathcal{Z}_{\mathrm{tgt}} - \underbrace{\sum_{i=0}^{K-1} \mathrm{D}_{\mathrm{KL}}\left(p_i \,\|\, p_{i+1}\right)}_{\text{bias}}$$

- Under perfect transitions, also equivalent to $\mathrm{var}(w^{(i)})$
- **Goal**: minimize sum of KL divergences

Analyzing AIS paths

- Approach: approximate the bias with a functional
- For linear schedules,

$$K \sum_{i=0}^{K-1} \mathrm{D}_{\mathrm{KL}}(p_i \| p_{i+1}) \xrightarrow{K \to \infty} \mathcal{F}(\gamma) \equiv \frac{1}{2} \int_0^1 \dot{\boldsymbol{\theta}}(\beta)^T \mathbf{G}_{\boldsymbol{\theta}}(\beta) \dot{\boldsymbol{\theta}}(\beta) \, \mathrm{d}\beta,$$

where $\mathbf{G}_{\boldsymbol{\theta}} \triangleq \underset{p_{\boldsymbol{\theta}}}{\mathrm{cov}}(\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}))$ denotes Fisher information

- Related to information geometry
- Same functional as for path sampling (Gelman and Meng, 1998)

Optimal schedules

- The cost under the optimal schedule is $\ell(\gamma)^2/2$, where

$$\ell(\gamma) = \int_0^1 \sqrt{\dot{\boldsymbol{\theta}}(\beta)^T \mathbf{G}_{\boldsymbol{\theta}}(\beta) \dot{\boldsymbol{\theta}}(\beta)} d\beta$$

is the path length on the Riemannian manifold with metric $\mathbf{G}_{\theta}$

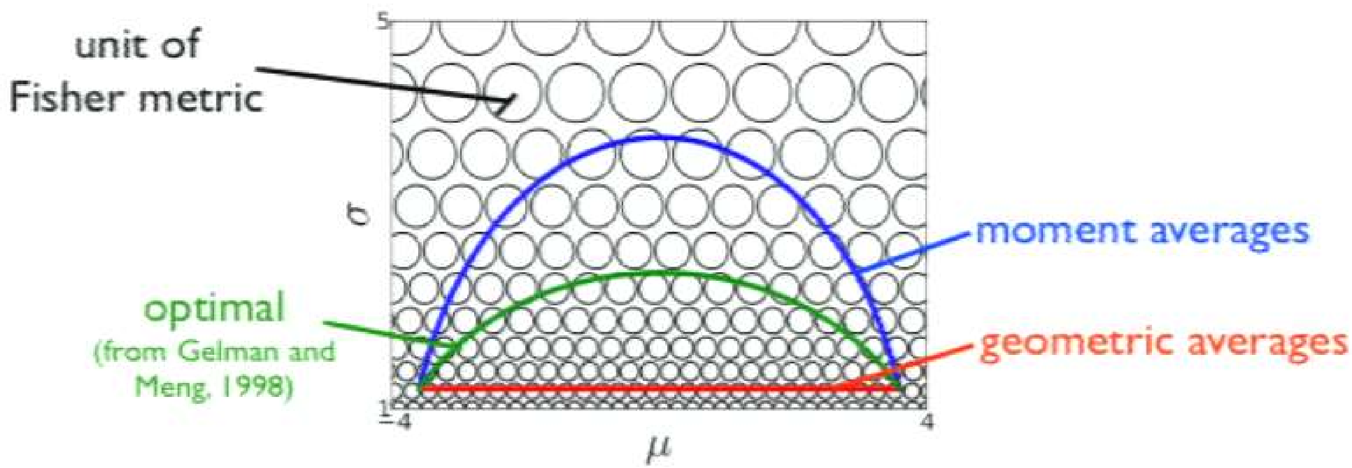
Optimal schedules

- The cost under the optimal schedule is $\ell(\gamma)^2/2$, where

$$\ell(\gamma) = \int_0^1 \sqrt{\dot{\boldsymbol{\theta}}(\beta)^T \mathbf{G}_{\boldsymbol{\theta}}(\beta) \dot{\boldsymbol{\theta}}(\beta)} d\beta$$

  is the path length on the Riemannian manifold with metric $\mathbf{G}_{\theta}$

- Example: annealing between univariate Gaussians



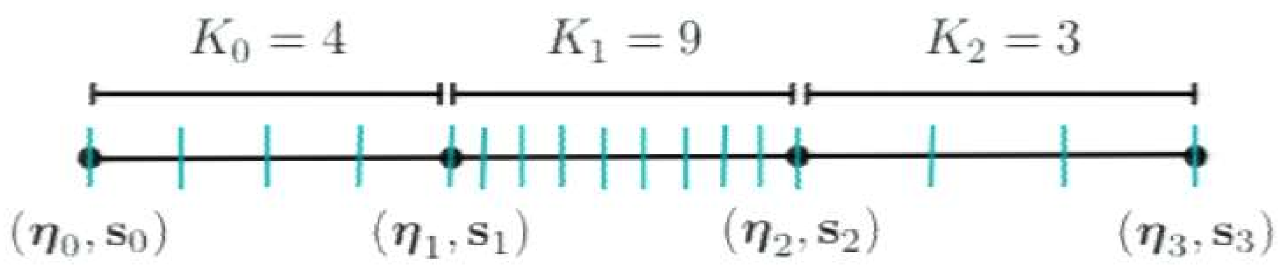
Optimal schedules

- Number of intermediate distributions needed to anneal between $\mathcal{N}(0, 1)$ and $\mathcal{N}(d, 1)$

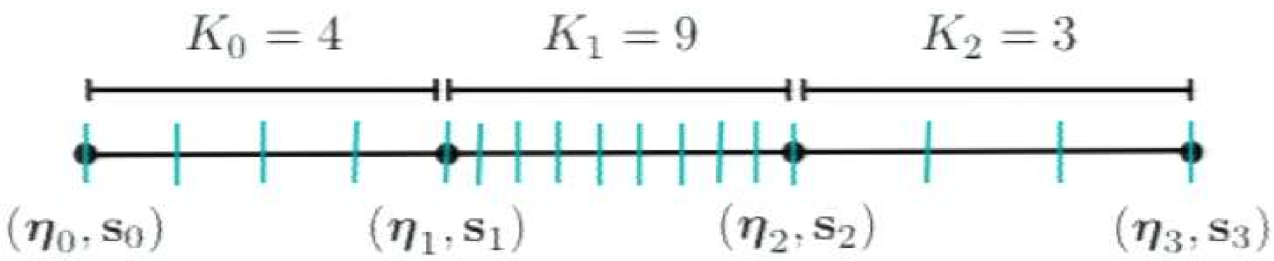
GA, linear schedule	$\mathcal{O}(d^2)$
GA, optimal schedule	$\mathcal{O}(d^2)$
MA, linear schedule	$\mathcal{O}(d^2)$
MA, optimal schedule	$\mathcal{O}((\log d)^2)$
Optimal path (Gelman and Meng, 1998)	$\mathcal{O}((\log d)^2)$

- MA within a constant factor of the optimal path

Optimal schedules



Optimal schedules

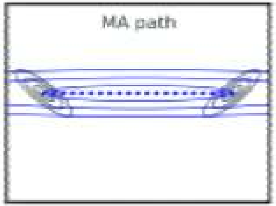


- Optimal piecewise linear schedule

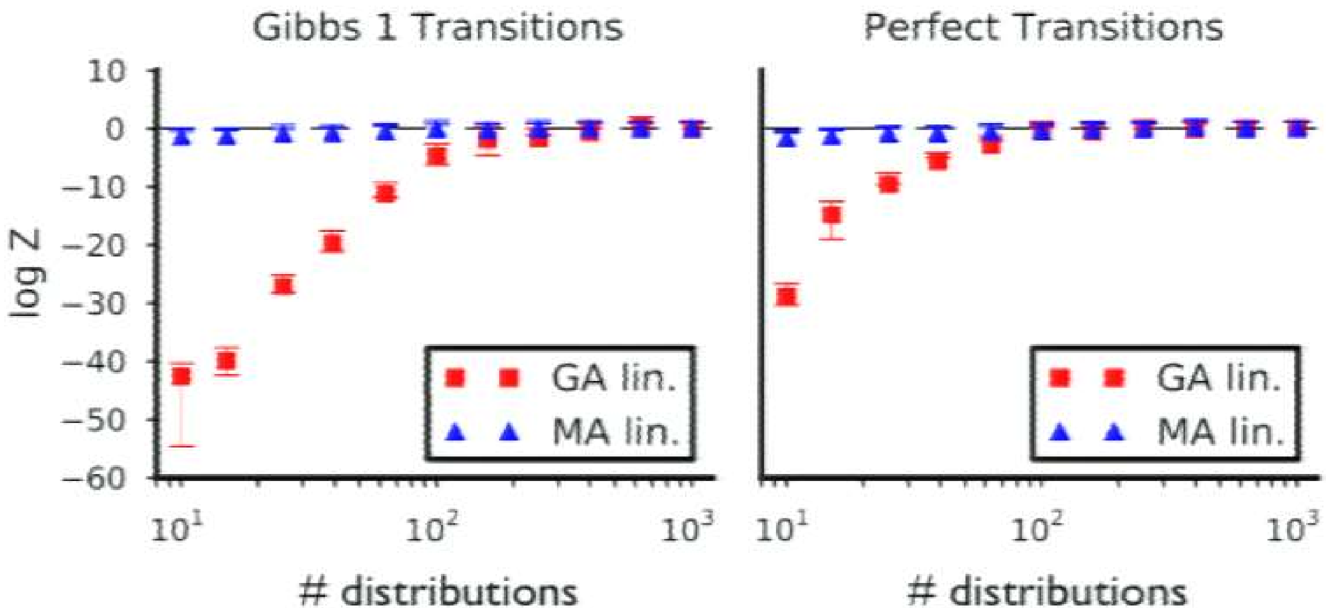
$$K_j \propto \sqrt{(\boldsymbol{\eta}_{j+1} - \boldsymbol{\eta}_j)^T (\mathbf{s}_{j+1} - \mathbf{s}_j)}$$

- **Caveat:** this assumes perfect transitions, and mixing effects are significant!

Experiments

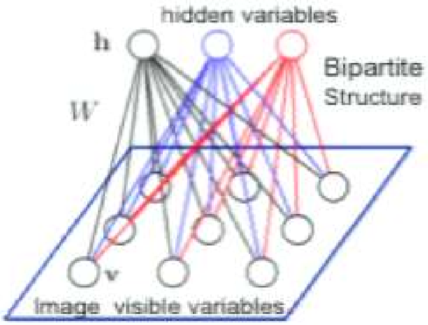


Multivariate Gaussians



Experiments

restricted Boltzmann machines



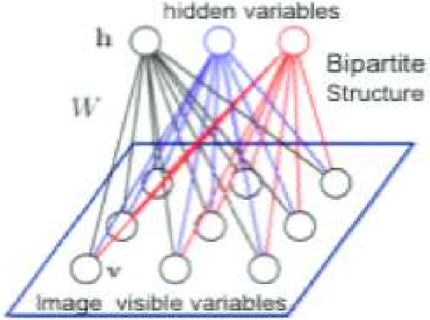
$$f(\mathbf{v},\mathbf{h}) = \exp\left(\mathbf{v}^T\mathbf{W}\mathbf{h} + \mathbf{v}^T\mathbf{c} + \mathbf{h}^T\mathbf{b}\right)$$

natural parameters: $\mathbf{W}, \mathbf{c}, \mathbf{b}$

moments: $\mathbb{E}[\mathbf{v}\mathbf{h}^T], \mathbb{E}[\mathbf{v}], \mathbb{E}[\mathbf{h}]$

Experiments

restricted Boltzmann machines



$$f(\mathbf{v}, \mathbf{h}) = \exp\left(\mathbf{v}^T \mathbf{W} \mathbf{h} + \mathbf{v}^T \mathbf{c} + \mathbf{h}^T \mathbf{b}\right)$$

natural parameters: $\mathbf{W}, \mathbf{c}, \mathbf{b}$

moments: $\mathbb{E}[\mathbf{v}\mathbf{h}^T], \mathbb{E}[\mathbf{v}], \mathbb{E}[\mathbf{h}]$

- Moment averaging:

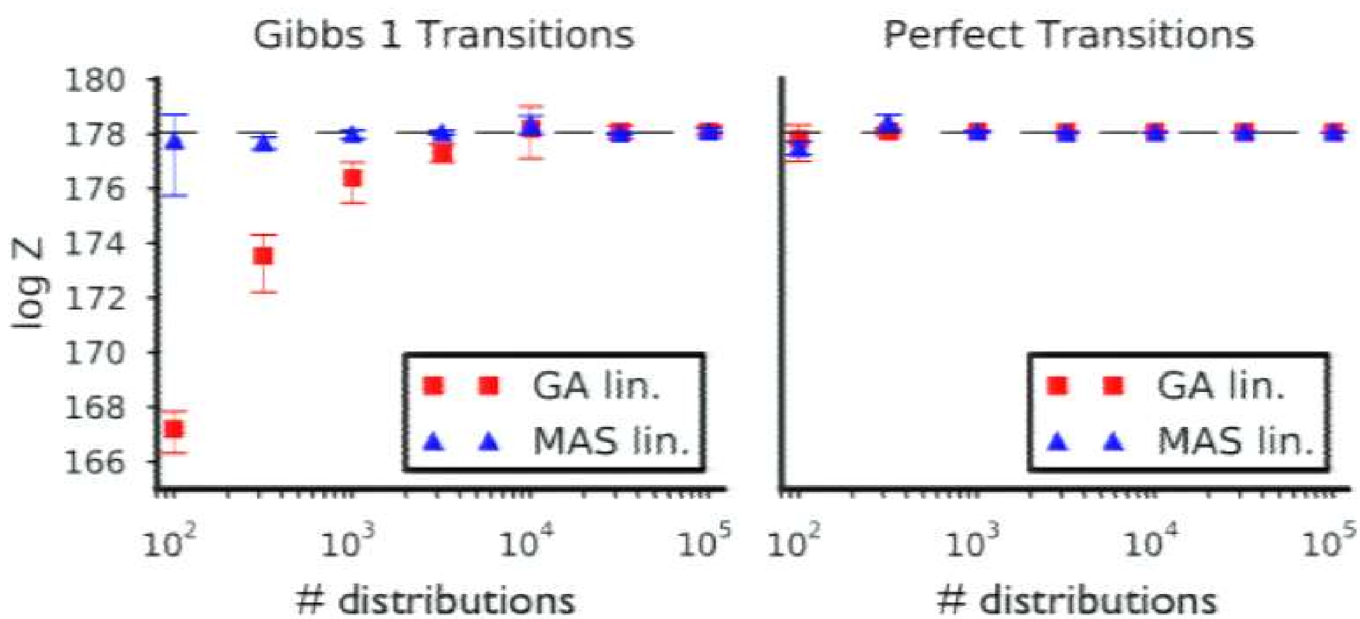
$$\overbrace{\mathbb{E}[\mathbf{v}\mathbf{h}^T]_\beta}^{\text{solve for natural parameters}} = (1-\beta)\mathbb{E}[\mathbf{v}\mathbf{h}^T]_{\text{init}} + \beta\, \overbrace{\mathbb{E}[\mathbf{v}\mathbf{h}^T]_{\text{tgt}}}^{\text{estimate moments}}$$

- Approximate with persistent contrastive divergence
- Solve for a few β values, interpolate with GA

Experiments

restricted Boltzmann machines

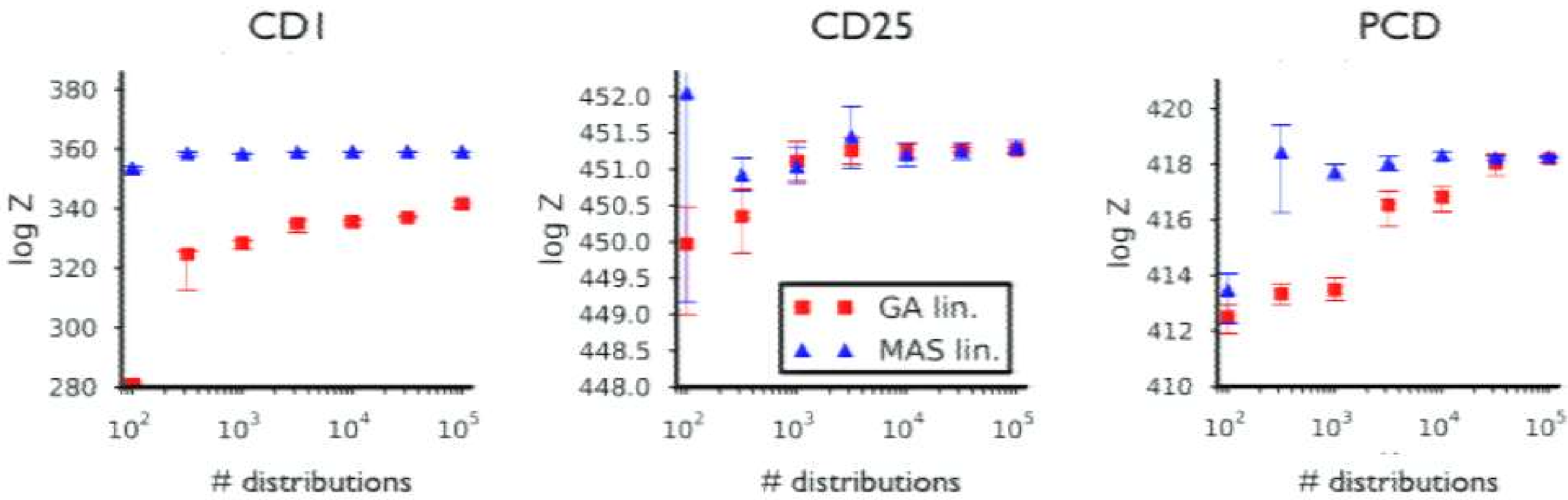
20 hidden units, trained on MNIST with PCD



Experiments

restricted Boltzmann machines

500 hidden units, trained on MNIST



Experiments

restricted Boltzmann machines

500 hidden units, trained on MNIST

beta = 0.00

geometric averages

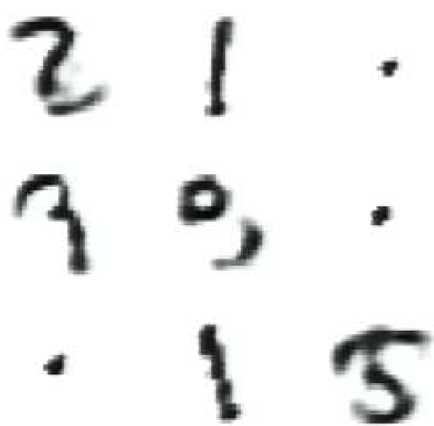
beta = 0.00

moment averages

Experiments

restricted Boltzmann machines

500 hidden units, trained on MNIST



beta = 1.00

geometric averages

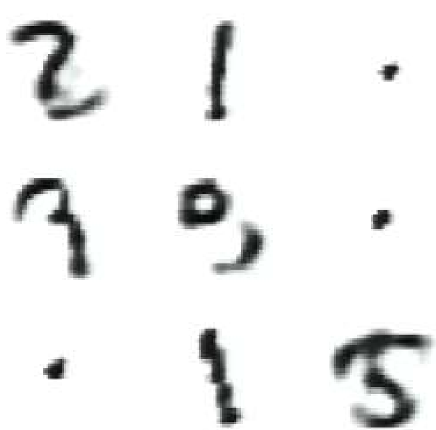
beta = 0.15

moment averages

Experiments

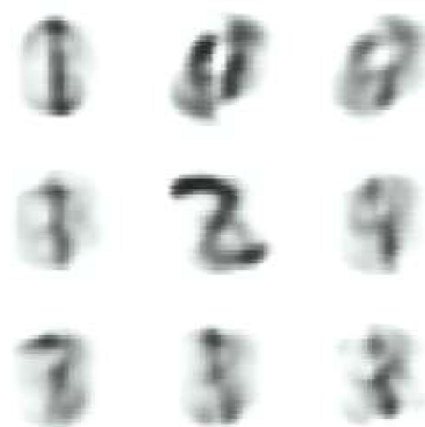
restricted Boltzmann machines

500 hidden units, trained on MNIST



beta = 1.00

geometric averages



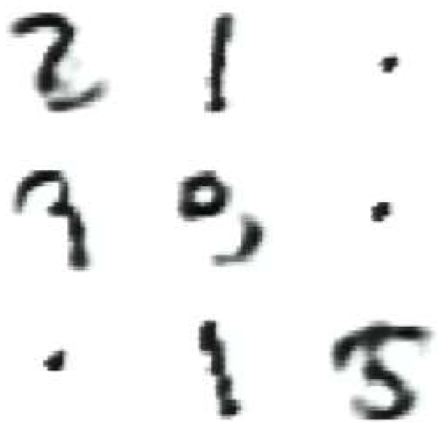
beta = 0.23

moment averages

Experiments

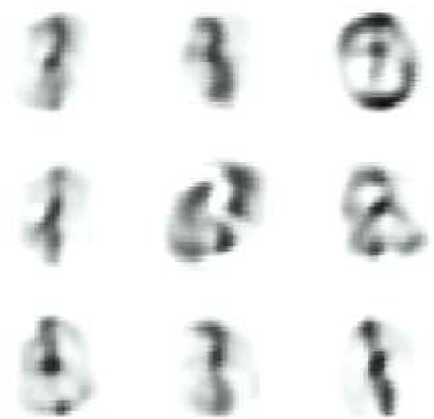
restricted Boltzmann machines

500 hidden units, trained on MNIST



beta = 1.00

geometric averages



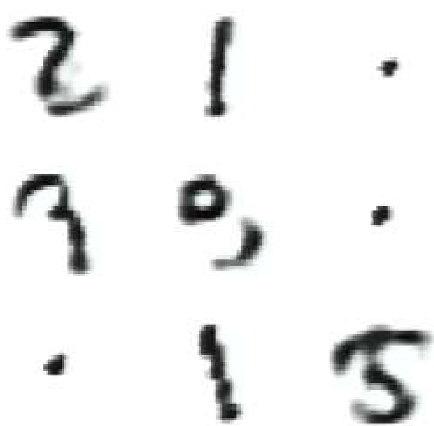
beta = 0.37

moment averages

Experiments

restricted Boltzmann machines

500 hidden units, trained on MNIST



beta = 1.00

geometric averages

beta = 0.55

moment averages

Conclusions

- The choice of path is a key design decision!
- Contributions
  - theoretical framework for analyzing annealing paths
  - novel path based on averaging moments
  - effective performance at estimating partition functions of RBMs
- Potentially relevant to any algorithm based on annealing paths
  - e.g. AIS, path sampling, thermodynamic integration, tempered transitions, parallel tempering, nested sampling, sequential Monte Carlo

Conclusions

- The choice of path is a key design decision!
- Contributions
  - theoretical framework for analyzing annealing paths
  - novel path based on averaging moments
  - effective performance at estimating partition functions of RBMs
- Potentially relevant to any algorithm based on annealing paths
  - e.g. AIS, path sampling, thermodynamic integration, tempered transitions, parallel tempering, nested sampling, sequential Monte Carlo

Poster Fri 13

NIPS Thanks Its Sponsors

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH · UNITED STATES AIR FORCE

amazon.com

Microsoft Research

Google

facebook

SKYTREE. THE MACHINE LEARNING COMPANY

TWO SIGMA

United Technologies Research Center

D E Shaw & Co

YAHOO! LABS

IBM Research

xerox

DRW TRADING GROUP

TOYOTA

millionshort

criteo

PDT PARTNERS

Springer Machine Learning Journal

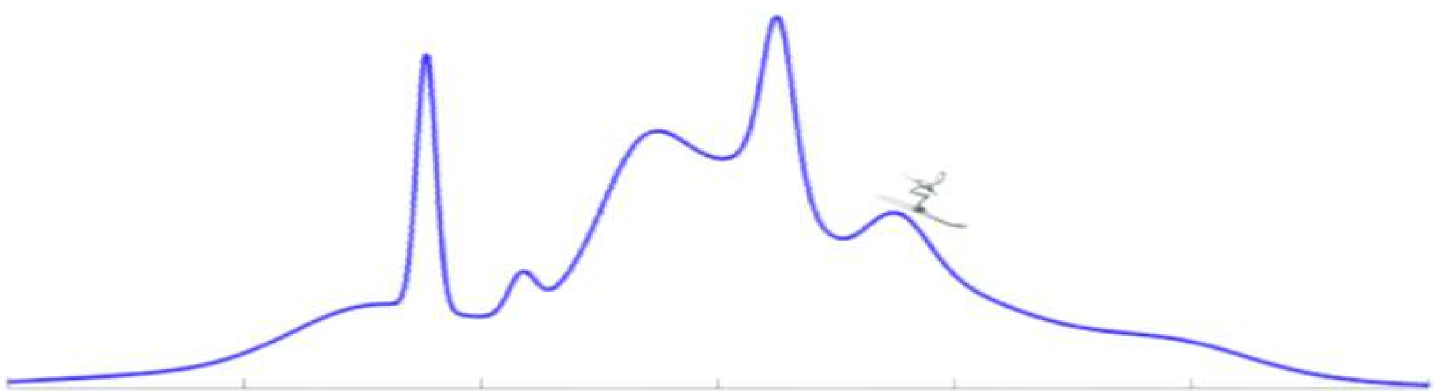
Disney Research

Dirichlet process mixture inconsistency for the number of components

Jeffrey W. Miller
and
Matthew T. Harrison

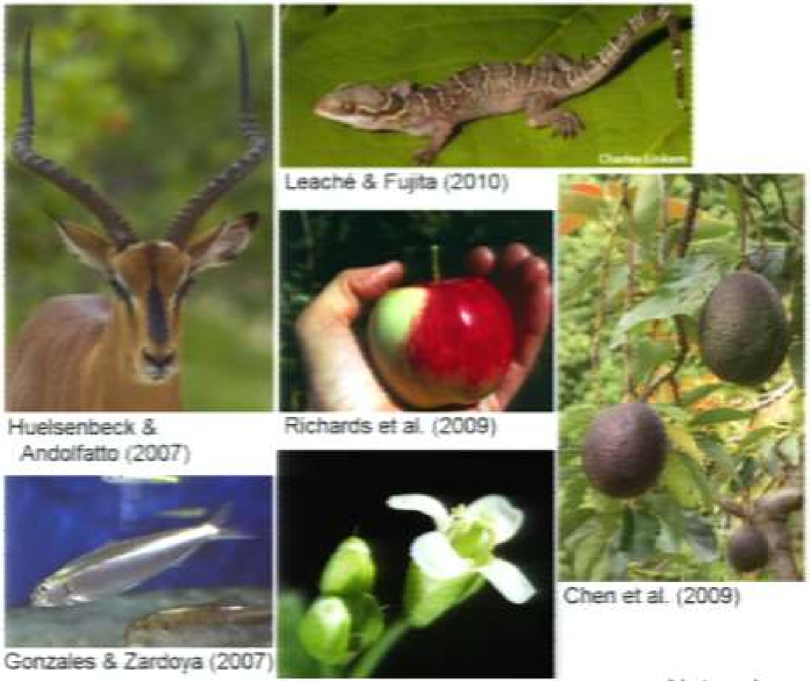
Brown University
Division of Applied Mathematics

NIPS 2013, Lake Tahoe



DPs are often used to infer the number of groups

Population structure



Huelsenbeck & Andolfatto (2007)

Leaché & Fujita (2010)

Richards et al. (2009)

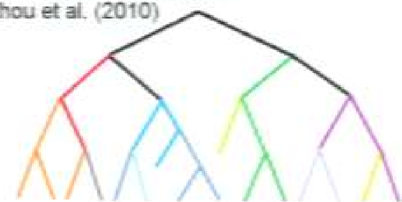
Chen et al. (2009)

Gonzales & Zardoya (2007)

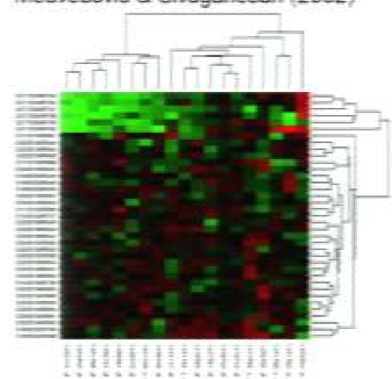
Fogelqvist et al. (2010)

Exchange rate modeling
Otranto & Gallo (2002)

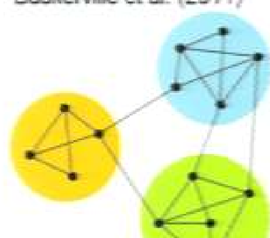
Heterotachy in phylogenetic trees
Lartillot & Philippe (2004)
Zhou et al. (2010)



Gene expression profiling
Medvedovic & Sivaganesan (2002)



Network communities
Baskerville et al. (2011)



Haplotype inference
Xing et al. (2006)

The DPM is great as a flexible prior on densities . . .

The DPM is great as a flexible prior on densities . . .

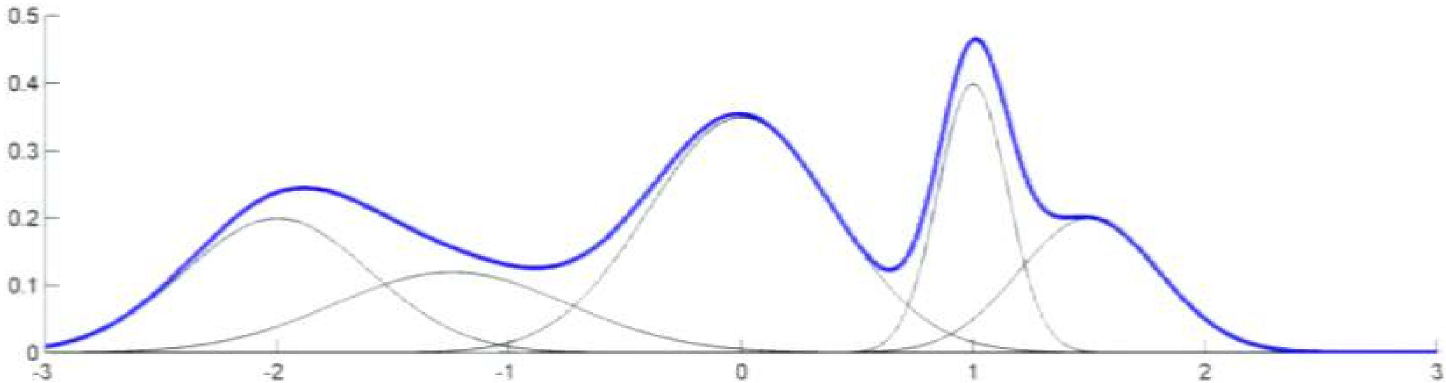
. . . what about for **estimating the number of groups**?

Finite mixture model

$$(\pi_1, \ldots, \pi_k) \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$$

$$\theta_1, \ldots, \theta_k \overset{\text{iid}}{\sim} H$$

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} f(x) = \sum_{i=1}^{k} \pi_i \, p_{\theta_i}(x)$$

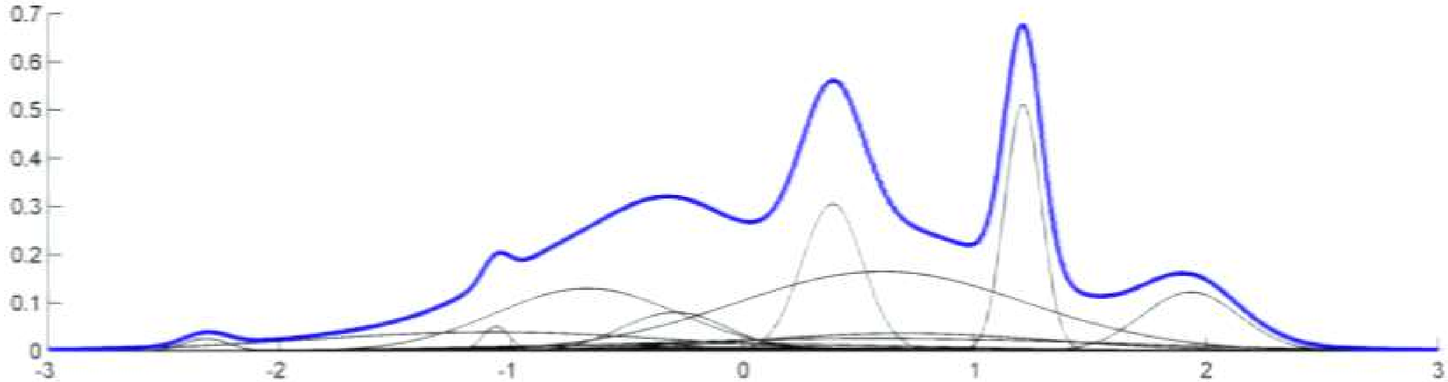


Dirichlet process mixture model

$$(\pi_1, \pi_2, \dots) \sim \text{Stick-breaking process}$$

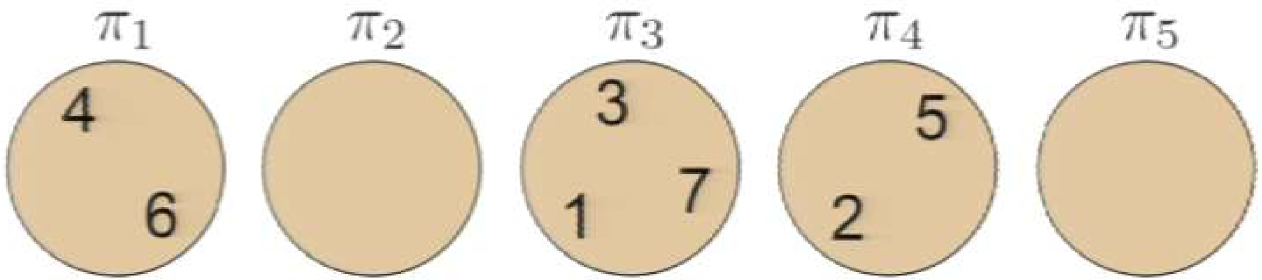
$$\theta_1, \theta_2, \dots \overset{\text{iid}}{\sim} H$$

$$X_1, \dots, X_n \overset{\text{iid}}{\sim} f(x) = \sum_{i=1}^{\infty} \pi_i \, p_{\theta_i}(x)$$



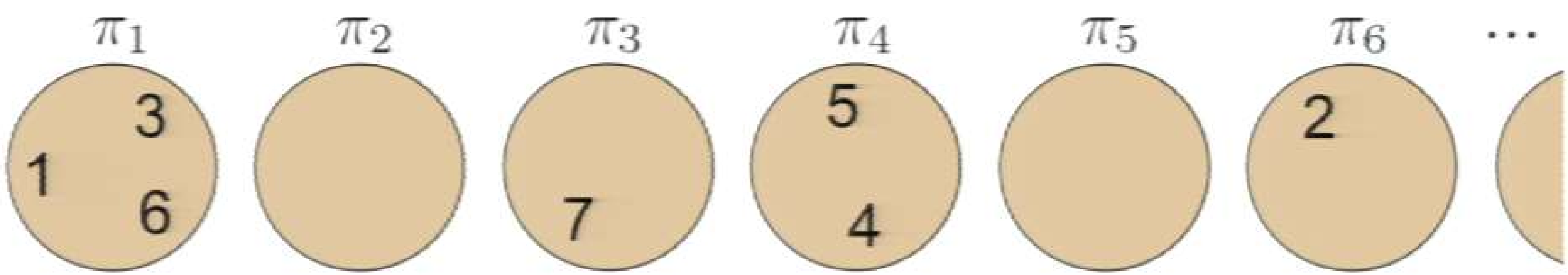
Ferguson (1983), Lo (1984), Sethuraman (1994),
West, Müller, and Escobar (1994), MacEachern (1994)

Finite mixture



5 tables (i.e. components)
3 occupied tables

Dirichlet process mixture



∞ tables (i.e. components)
4 occupied tables

What if we use a DPM on data from finite mixture?

It is known that in many cases the posterior concentrates at the true density f_0,

$$P(\|f - f_0\|_{L_1} < \varepsilon \mid X_{1:n}) \xrightarrow[n\to\infty]{} 1 \;\; \forall \varepsilon > 0,$$

(often at essentially the minimax-optimal rate), for *any* sufficiently regular f_0.
(Contributions by: Ghosal, van der Vaart, Scricciolo, Lijoi, Prünster, Walker, James, Tokdar, Dunson, Bhattacharya, Wu, Ghosh, Ramamoorthi, Ishwaran, and others.)

What if we use a DPM on data from finite mixture?

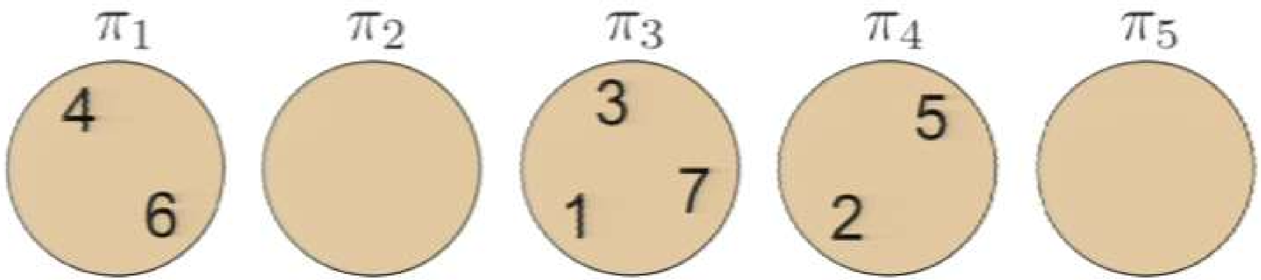
It is known that in many cases the posterior concentrates at the true density f_0,

$$P(\|f - f_0\|_{L_1} < \varepsilon \mid X_{1:n}) \xrightarrow[n\to\infty]{} 1 \ \forall \varepsilon > 0,$$

(often at essentially the minimax-optimal rate), for *any* sufficiently regular f_0.
(Contributions by: Ghosal, van der Vaart, Scricciolo, Lijoi, Prünster, Walker, James, Tokdar, Dunson, Bhattacharya, Wu, Ghosh, Ramamoorthi, Ishwaran, and others.)

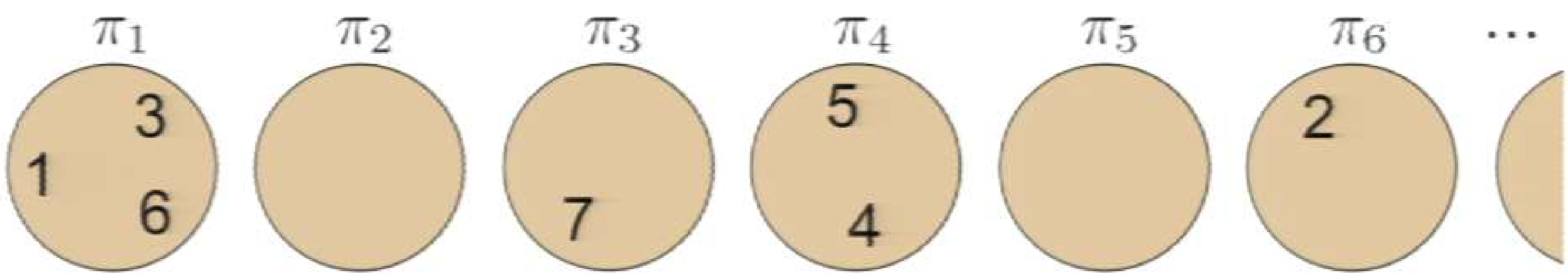
In fact, the posterior on the mixing distribution concentrates (in Wasserstein distance) at the true mixing distribution (Nguyen, 2013).

Finite mixture



5 tables (i.e. components)
3 occupied tables

Dirichlet process mixture



∞ tables (i.e. components)
4 occupied tables

What if we use a DPM on data from finite mixture?

It is known that in many cases the posterior concentrates at the true density f_0,

$$P(\|f - f_0\|_{L_1} < \varepsilon \mid X_{1:n}) \xrightarrow[n\to\infty]{} 1 \;\; \forall \varepsilon > 0,$$

(often at essentially the minimax-optimal rate), for *any* sufficiently regular f_0. (Contributions by: Ghosal, van der Vaart, Scricciolo, Lijoi, Prünster, Walker, James, Tokdar, Dunson, Bhattacharya, Wu, Ghosh, Ramamoorthi, Ishwaran, and others.)

In fact, the posterior on the mixing distribution concentrates (in Wasserstein distance) at the true mixing distribution (Nguyen, 2013).

Does the posterior on the number of occupied tables concentrate at the true number of components? i.e.

$$P(\#\text{occupied} = k_0 \mid X_{1:n}) \xrightarrow[n\to\infty]{?} 1$$

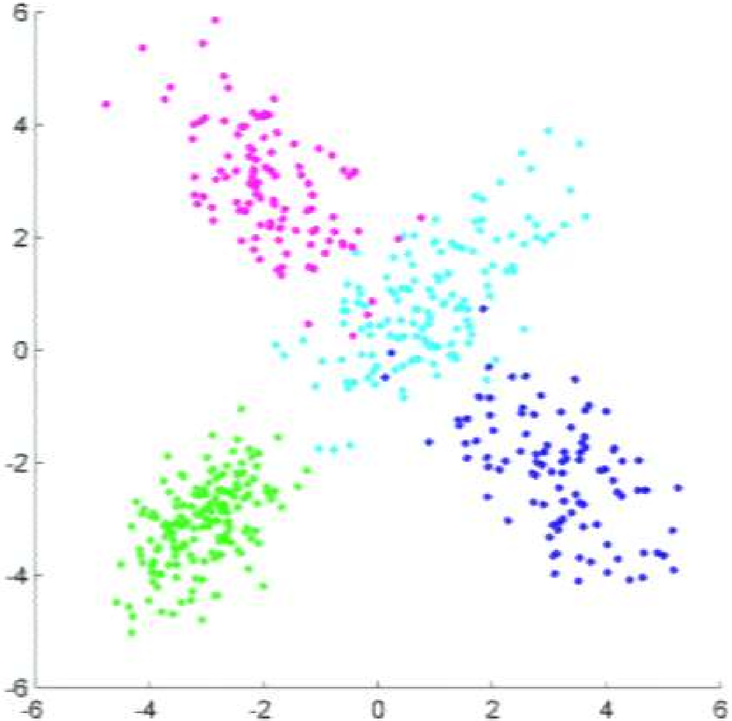
Outline

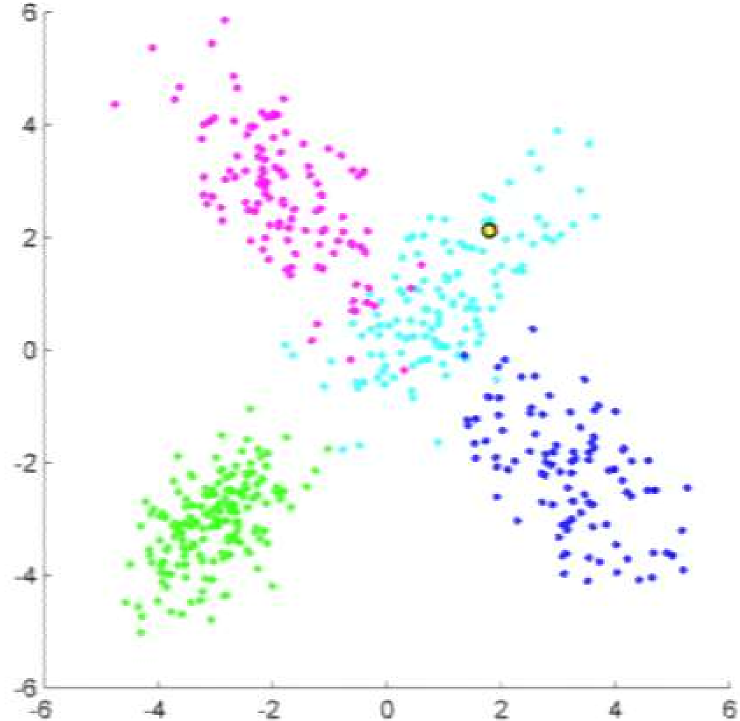
1. Empirical evidence
2. Theoretical results
3. Intuition

Tiny extra clusters often appear in posterior samples.

Empirically, this is well-known (e.g. West, Müller, and Escobar, 1994).

Bivariate Gaussian mixture with 4 components

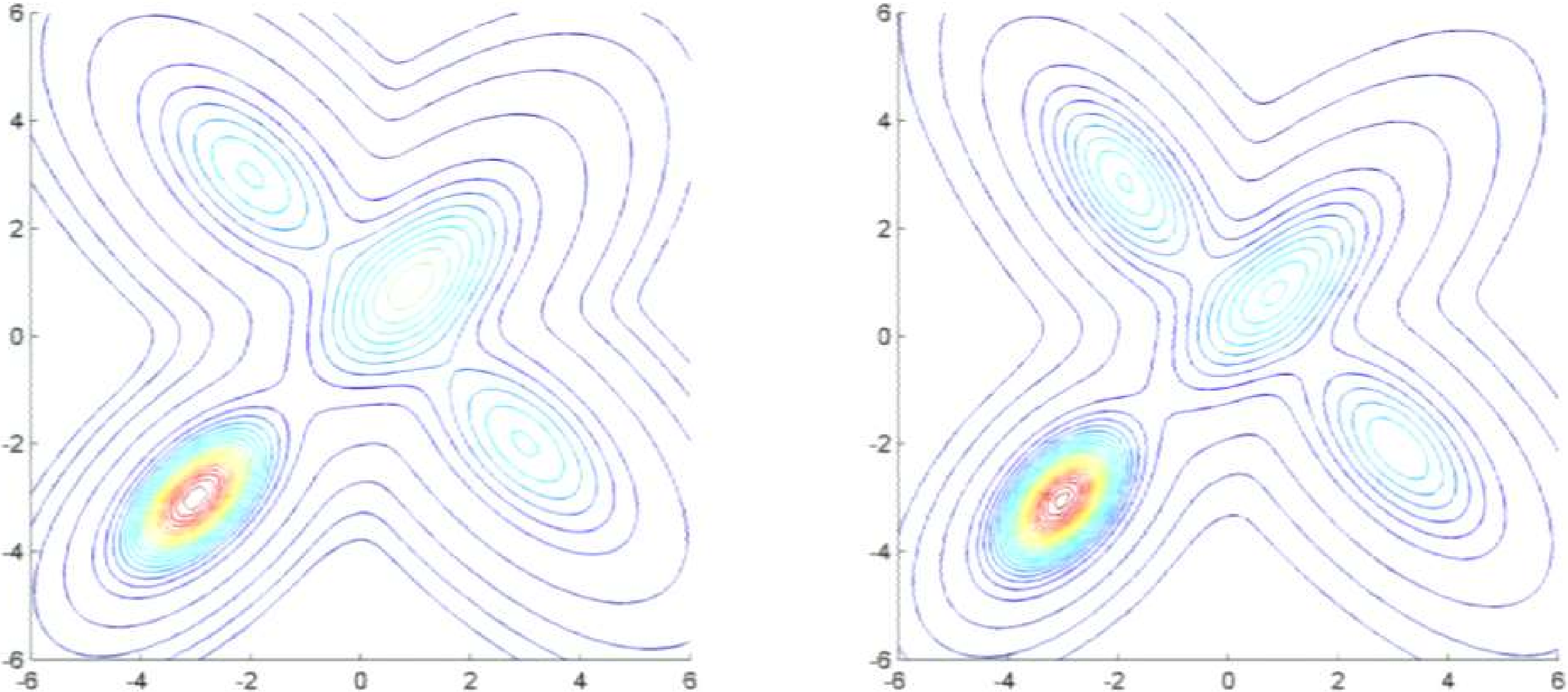




Tiny extra clusters often appear in posterior samples.

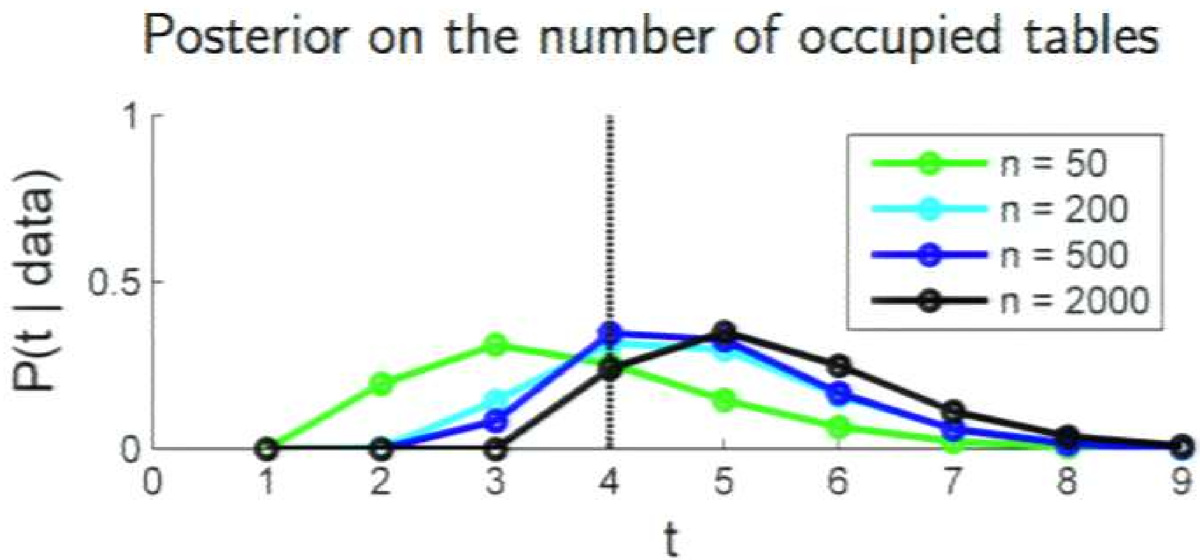
Bivariate Gaussian mixture with 4 components

True density | Posterior predictive density



These tiny clusters have negligible impact on density estimates ...

Bivariate Gaussian mixture with 4 components



. . . but they do affect the posterior on the number of occupied tables.

Theoretical results

Theorem (M. & Harrison, 2013)

Under mild regularity conditions, if $X_1, X_2, \ldots$ are i.i.d. from a finite mixture with k_0 components, then the DPM posterior on the number of occupied tables T_n satisfies

$$\limsup_{n \to \infty} P(T_n = k_0 \mid X_1, \ldots, X_n) < 1$$

with probability 1.

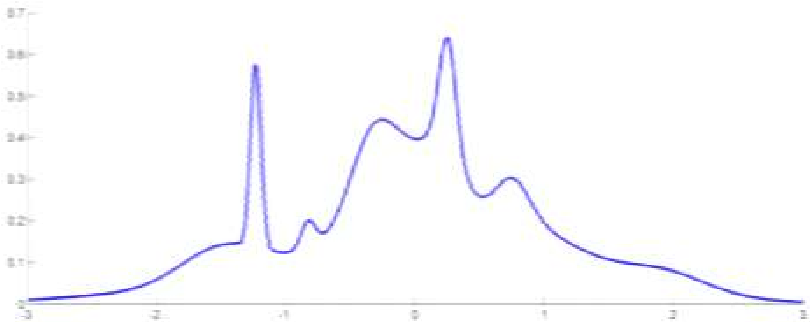
- This implies inconsistency.
- We assume the concentration parameter α is fixed.
- This generalizes to Pitman–Yor process mixtures.
- See Miller & Harrison (2013) arXiv:1309.0024 for details.

This implies inconsistency of Dirichlet process mixtures over:

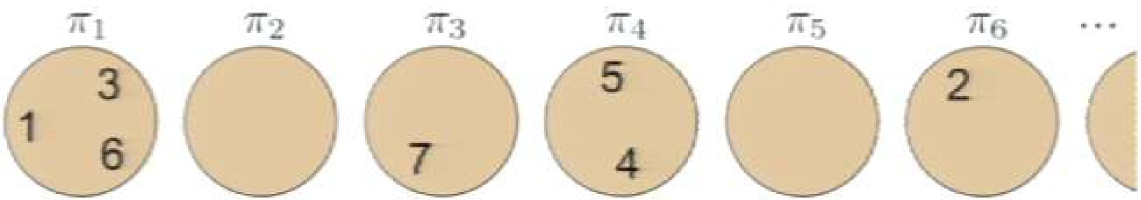
1. a large class of continuous exponential families, including
   - multivariate Gaussian
   - Exponential
   - Gamma
   - Log-Normal
   - Weibull with fixed shape
2. essentially any discrete family, including
   - Poisson
   - Geometric
   - Negative Binomial
   - Binomial
   - Multinomial
   - (and many more)

To be clear: It's fine to use DPMs ...

1. as a flexible prior on densities
   (viewing the latent variables as nuisance parameters)



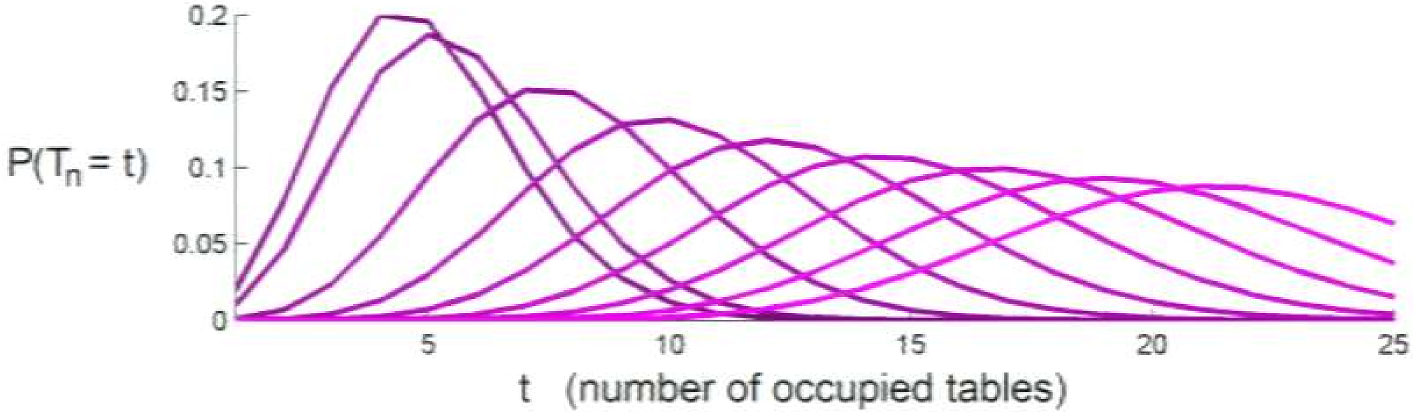
2. or if the data-generating process is well-modeled by a DPM
   (and in particular, is not a finite mixture!)



Intuition

The wrong intuition

It is tempting to think that the prior on the number of occupied tables is the culprit, since it is diverging as $n \to \infty$.



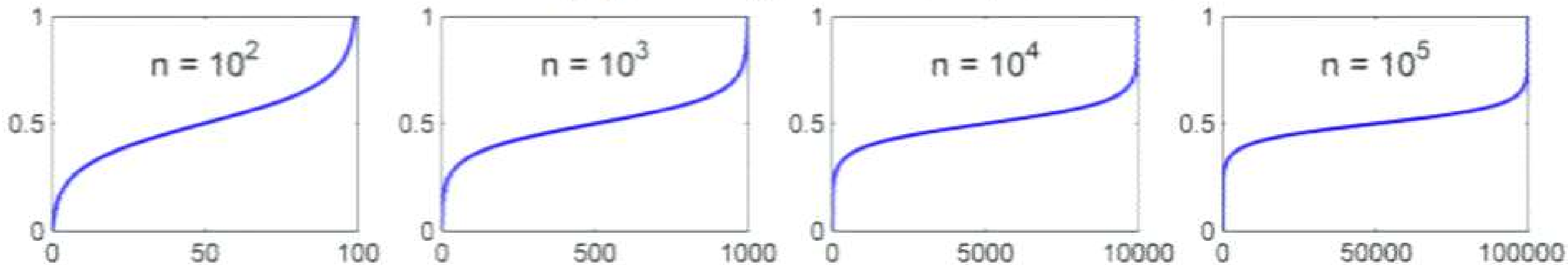
However, this is not the fundamental reason why inconsistency occurs.

The right intuition

Given that there are t occupied tables, the conditional distribution of their sizes $n_1, \ldots, n_t$ is

$$P(n_1, \ldots, n_t \mid T_n = t) \propto n_1^{-1} \cdots n_t^{-1} I(\sum n_i = n).$$

CDF of n_1 given $T_n = 2$ occupied tables

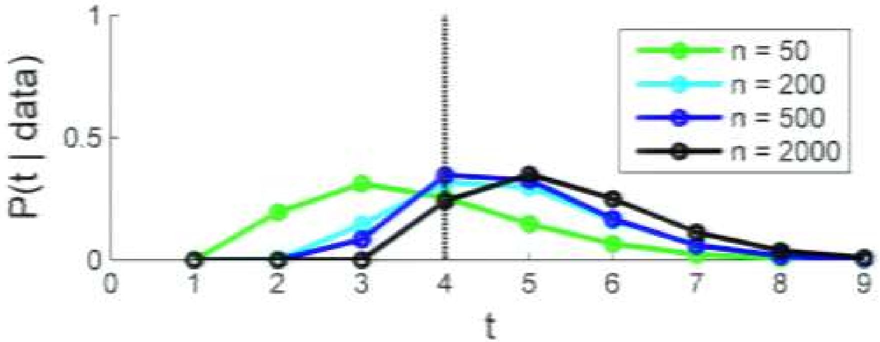


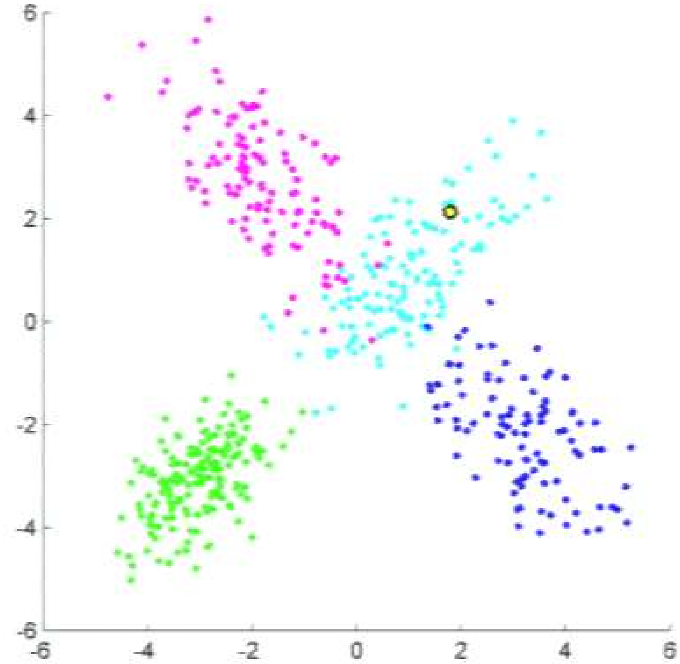
Key observation

As n grows, this becomes concentrated in the "corners". In other words, the DPM really likes to have one or more tables with very few customers.

The DPM really likes to have one or more tables with very few customers.

This explains the tiny extra clusters, since (it turns out) they do not significantly reduce the likelihood.





Solutions?

What if we ...

- put a prior on the concentration parameter?
- ignore tables with very few customers? (busy waiter strategy)
- put a prior on the number of components?

  This works in principle (Nobile, 1994), but ...

beware of misspecification.

Summary

The DPM posterior on the number of occupied tables should not be used to estimate the number of components in a finite mixture.

Dirichlet process mixture inconsistency for the number of components

Jeffrey W. Miller
and
Matthew T. Harrison

Brown University
Division of Applied Mathematics

Poster: Fri37

NIPS Thanks Its Sponsors

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
UNITED STATES AIR FORCE

amazon.com

Microsoft Research

Google

facebook

SKYTREE
THE MACHINE LEARNING COMPANY

TWO SIGMA

United Technologies Research Center

D E Shaw & Co

xerox

YAHOO! LABS

IBM Research

DRW TRADING GROUP

TOYOTA

millionshort

criteo

PDT PARTNERS

Springer
Machine Learning Journal

Disney Research

Approximate Bayesian Image Interpretation via Generative Probabilistic Graphics Programs

Vikash K. Mansinghka[*1,2], Tejas D. Kulkarni[*1,2] Yura N. Perov[3] Joshua B. Tenenbaum[1,2]

[1]Computer Science & Artificial Intelligence Laboratory

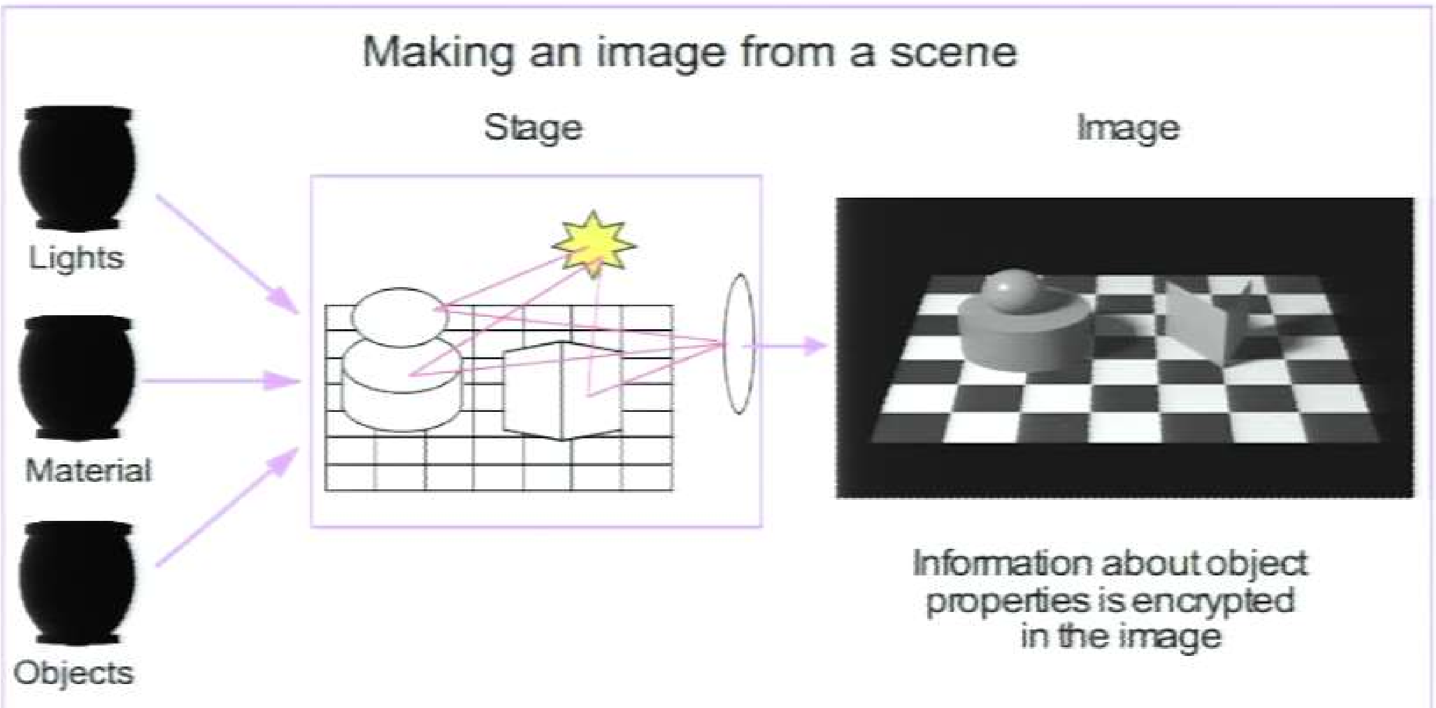
[2]Department of Brain and Cognitive Sciences

Massachusetts Institute of Technology

[3]Institute of Mathematics and Computer Science

Siberian Federal University

Vision as Inverse Graphics



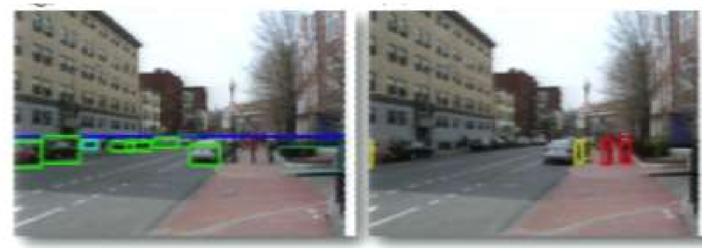
Kersten, NIPS 1998 Tutorial on Computational Vision

“Taking Inverse Graphics Seriously”

“Taking Inverse Graphics Seriously”

Combining bottom-up classifiers, search and 3D geometry

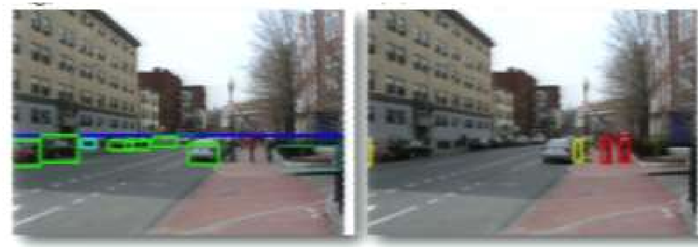
(Gupta, Efros and Hebert 2010)



(Hoeim, Efros and Hebert 2006)

“Taking Inverse Graphics Seriously”

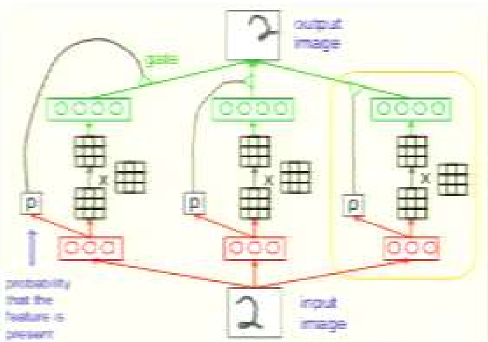
Combining bottom-up classifiers, search and 3D geometry

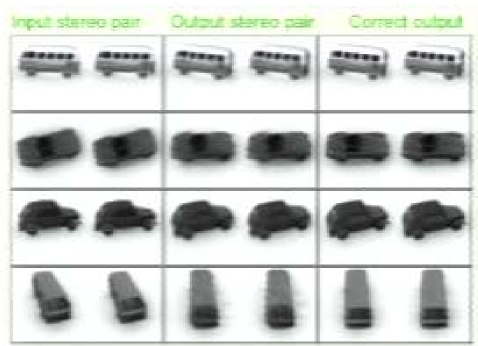


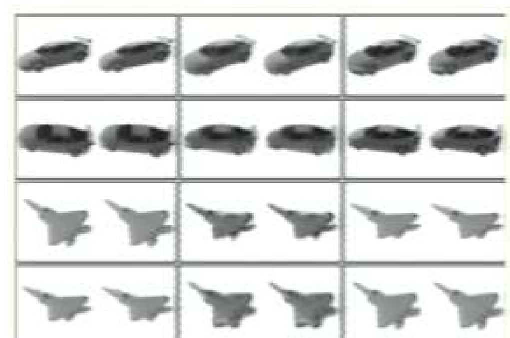
(Gupta, Efros and Hebert 2010)

(Hoeim, Efros and Hebert 2006)

Learning transforming autoencoders







(Hinton, Krizhevsky and Wang, 2011)

Generative Probabilistic Graphics Programming: Taking Inverse Graphics *Literally*

Generative Probabilistic Graphics Programming: Taking Inverse Graphics *Literally*

- **Direct formulations of approximately Bayesian inverse graphics are possible, given:**

    1. Generative models written as probabilistic graphics programs in Church/Venture

    2. Automatic, general-purpose samplers for inference; no custom inference code needed

    3. Approximate comparison of rendering and image data: a variation on ABC

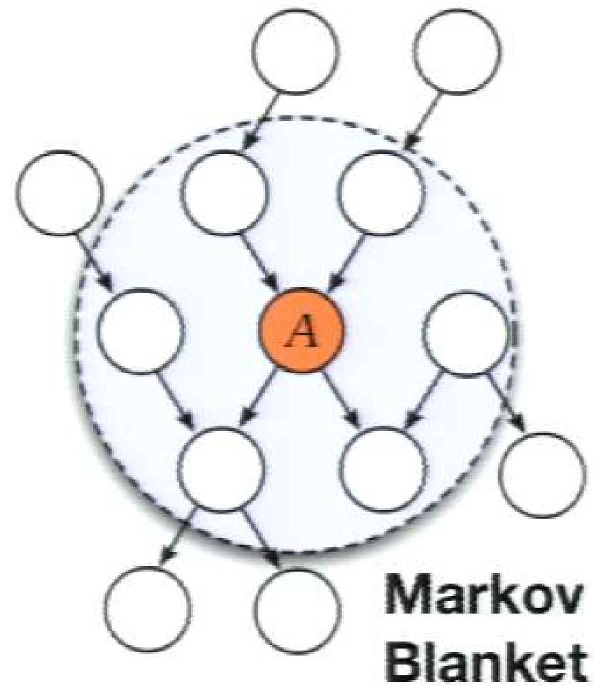
    4. Bayesian relaxations, to adaptively smooth the energy landscape

Generative Probabilistic Graphics Programming: Taking Inverse Graphics *Literally*

- **Direct formulations of approximately Bayesian inverse graphics are possible, given:**
  1. Generative models written as probabilistic graphics programs in Church/Venture
  2. Automatic, general-purpose samplers for inference; no custom inference code needed
  3. Approximate comparison of rendering and image data: a variation on ABC
  4. Bayesian relaxations, to adaptively smooth the energy landscape
- **Empirical demonstrations:**
  1. 2D: obscured digits + letters
  2. 3D: road scenes

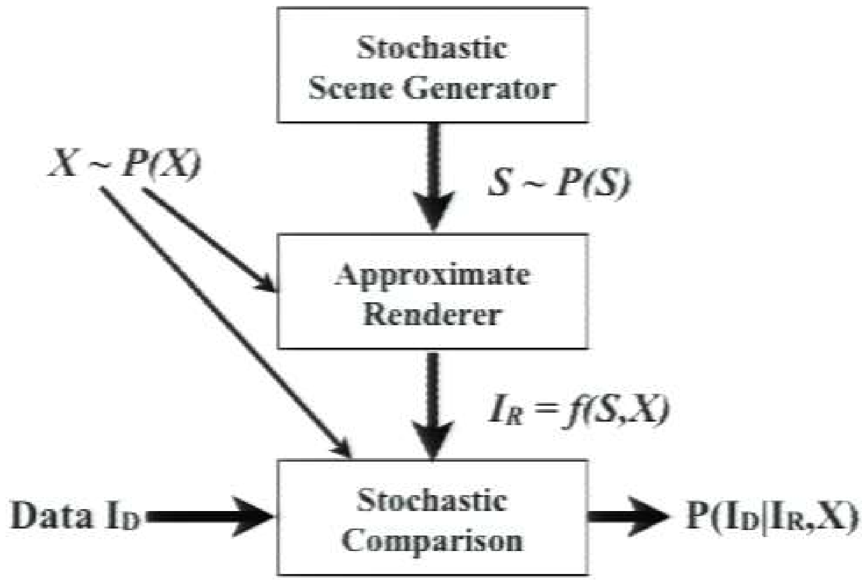
Probabilistic Programming with Church and Venture

```
ASSUME size        (uniform 0 1)
ASSUME pos_x       (uniform 0 1)
ASSUME pos_y       (uniform 0 1)

ASSUME rotation_x (uniform 0 180)
ASSUME rotation_y (uniform 0 180)
ASSUME rotation_z (uniform 0 180)

ASSUME image       (render_wire_cube size pos_x ...)

ASSUME blur_bw     (gamma 1 1)
ASSUME sigsq       (gamma 1 1)

ASSUME blurred     (gaussian_blur image blur_bw)

ASSUME data        (load_image "cube.png")
OBSERVE (multivariate_normal blurred sigsq) data
```

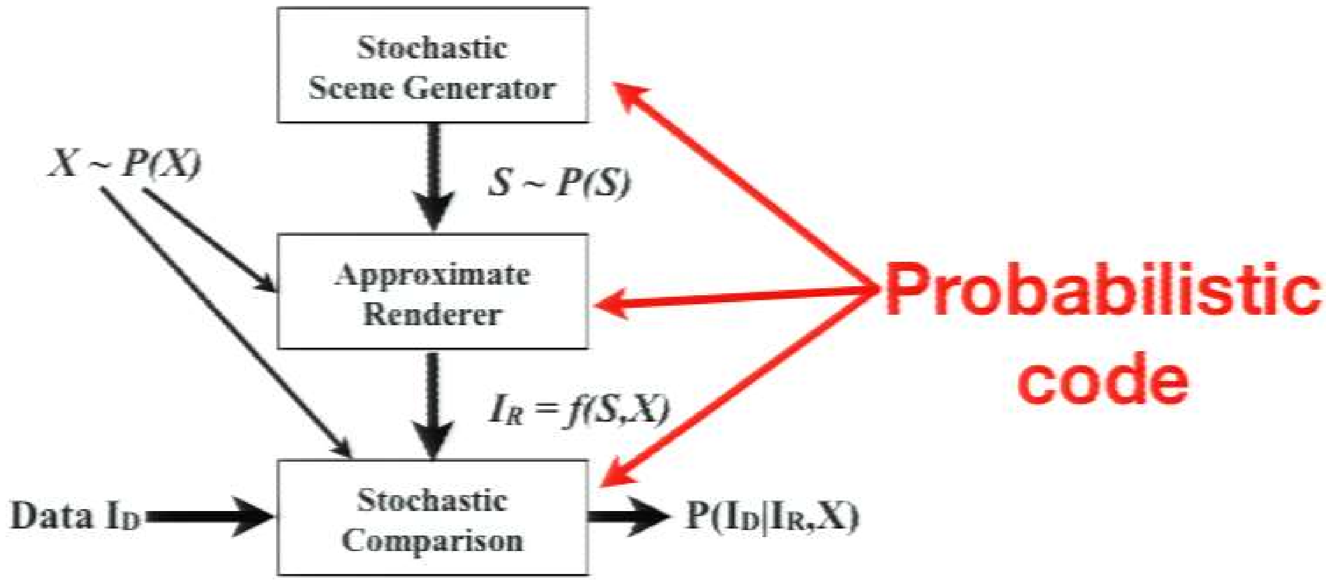


Probabilistic code in Venture, (Mansinghka, Selsam and Perov, in prep)
a new Turing-complete platform descended from Church (Goodman*, Mansinghka*, Roy et al., 2008)

Generative Probabilistic Graphics Programming: Taking Inverse Graphics *Literally*



Generative Probabilistic Graphics Programming: Taking Inverse Graphics *Literally*



$$P(S|I_D) \propto \int P(S)P(X)\delta_{f(S,X)}(I_R)P(I_D|I_R,X)dX$$

Automatic, general-purpose samplers for inference:

$$\alpha_{MH}((S,X) \to (S',X')) \;=\; min(1, \frac{P(I_D|f(S',X'),X')P(S')P(X')q((S',X') \to (S,X))}{P(I_D|f(S,X),X)P(S)P(X)q((S,X) \to (S',X'))})$$

GPGP Illustration: Reading Obscured Text

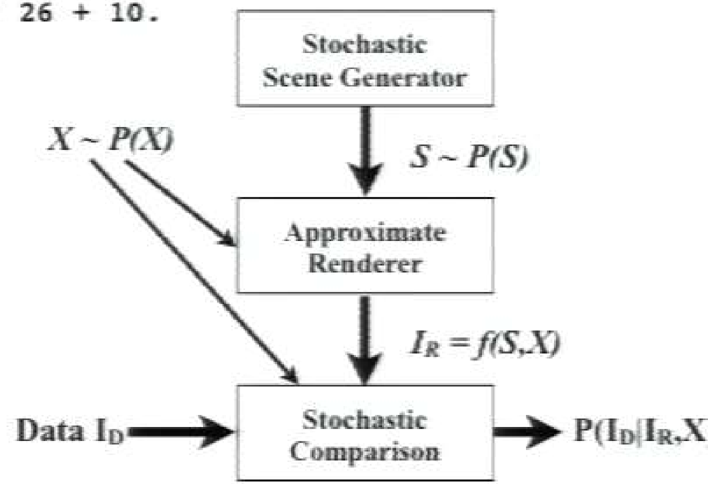
GPGP Illustration: Reading Obscured Text

```
ASSUME is_present (mem (lambda (id) (bernoulli 0.5)))
ASSUME pos_x (mem (lambda (id) (uniform_discrete 0 200)))
ASSUME pos_y (mem (lambda (id) (uniform_discrete 0 200)))
ASSUME size_x (mem (lambda (id) (uniform_discrete 0 100)))
ASSUME size_y (mem (lambda (id) (uniform_discrete 0 100)))
ASSUME rotation (mem (lambda (id) (uniform_continuous -20.0 20.0)))
ASSUME glyph (mem (lambda (id) (uniform_discrete 0 35))) // 26 + 10.

ASSUME blur (mem (lambda (id) (* 7 (beta 1 2))))
ASSUME global_blur (* 7 (beta 1 2))
ASSUME data_blur (* 7 (beta 1 2))
ASSUME epsilon (gamma 1 1)

ASSUME image (render_surfaces max-num-glyphs global_blur
(pos_x 1) (pos_y 1) (glyph 1) (size_x 1) (size_y 1)
(rotation 1) (blur 1)  (is_present 1)
(pos_x 2) (pos_y 2) (glyph 2) (size_x 2) (size_y 2)
(rotation 2) (blur 2) (is_present 2)
... (is_present 10))

ASSUME data (load_image "captcha_1.png" data_blur)
OBSERVE (incorporate_stochastic_likelihood data image epsilon) True
```

Probabilistic code in Venture, (Mansinghka, Selsam and Perov, in prep)
a new Turing-complete platform descended from Church (Goodman*, Mansinghka*, Roy et al., 2008)

GPGP Illustration: Reading Obscured Text

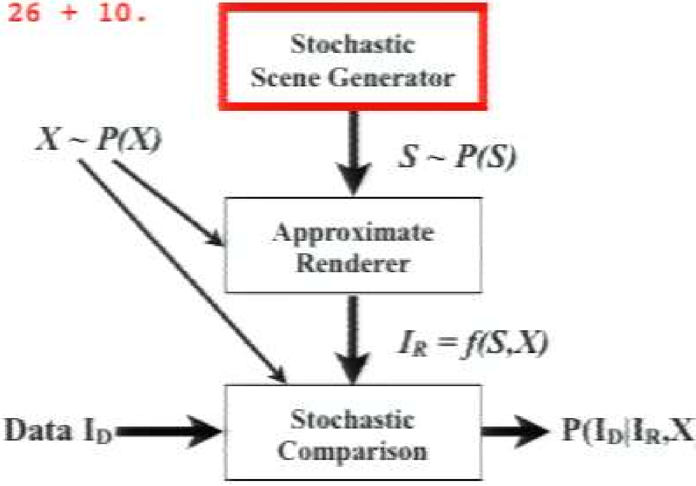
```
ASSUME is_present (mem (lambda (id) (bernoulli 0.5)))
ASSUME pos_x (mem (lambda (id) (uniform_discrete 0 200)))
ASSUME pos_y (mem (lambda (id) (uniform_discrete 0 200)))
ASSUME size_x (mem (lambda (id) (uniform_discrete 0 100)))
ASSUME size_y (mem (lambda (id) (uniform_discrete 0 100)))
ASSUME rotation (mem (lambda (id) (uniform_continuous -20.0 20.0)))
ASSUME glyph (mem (lambda (id) (uniform_discrete 0 35))) // 26 + 10.

ASSUME blur (mem (lambda (id) (* 7 (beta 1 2))))
ASSUME global_blur (* 7 (beta 1 2))
ASSUME data_blur (* 7 (beta 1 2))
ASSUME epsilon (gamma 1 1)

ASSUME image (render_surfaces max-num-glyphs global_blur
(pos_x 1) (pos_y 1) (glyph 1) (size_x 1) (size_y 1)
(rotation 1) (blur 1)  (is_present 1)
(pos_x 2) (pos_y 2) (glyph 2) (size_x 2) (size_y 2)
(rotation 2) (blur 2) (is_present 2)
... (is_present 10))

ASSUME data (load_image "captcha_1.png" data_blur)
OBSERVE (incorporate_stochastic_likelihood data image epsilon) True
```



Probabilistic code in Venture, (Mansinghka, Selsam and Perov, in prep)
a new Turing-complete platform descended from Church (Goodman*, Mansinghka*, Roy et al., 2008)

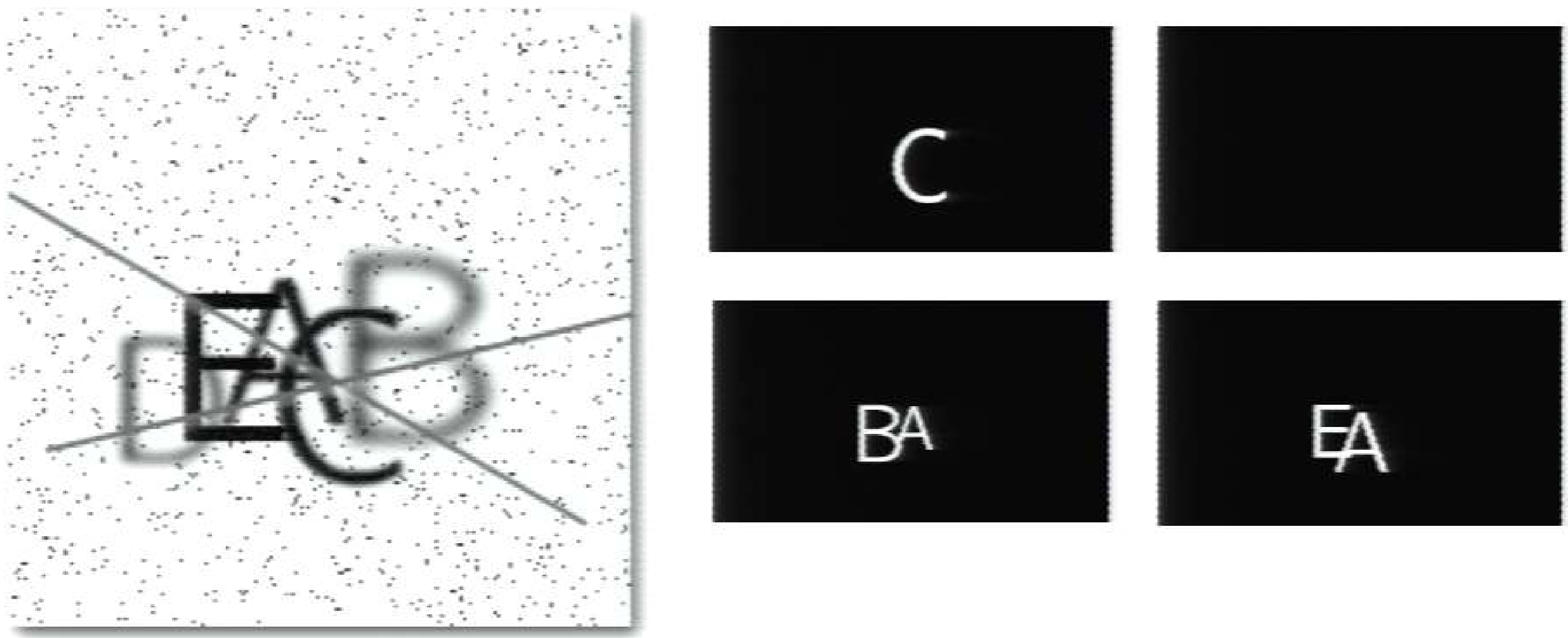
GPGP Illustration: Reading Obscured Text

```
ASSUME is_present (mem (lambda (id) (bernoulli 0.5)))
ASSUME pos_x (mem (lambda (id) (uniform_discrete 0 200)))
ASSUME pos_y (mem (lambda (id) (uniform_discrete 0 200)))
ASSUME size_x (mem (lambda (id) (uniform_discrete 0 100)))
ASSUME size_y (mem (lambda (id) (uniform_discrete 0 100)))
ASSUME rotation (mem (lambda (id) (uniform_continuous -20.0 20.0)))
ASSUME glyph (mem (lambda (id) (uniform_discrete 0 35))) // 26 + 10.

ASSUME blur (mem (lambda (id) (* 7 (beta 1 2))))
ASSUME global_blur (* 7 (beta 1 2))
ASSUME data_blur (* 7 (beta 1 2))
ASSUME epsilon (gamma 1 1)

ASSUME image (render_surfaces max-num-glyphs global_blur
(pos_x 1) (pos_y 1) (glyph 1) (size_x 1) (size_y 1)
(rotation 1) (blur 1)  (is_present 1)
(pos_x 2) (pos_y 2) (glyph 2) (size_x 2) (size_y 2)
(rotation 2) (blur 2) (is_present 2)
... (is_present 10))

ASSUME data (load_image "captcha_1.png" data_blur)
OBSERVE (incorporate_stochastic_likelihood data image epsilon) True
```



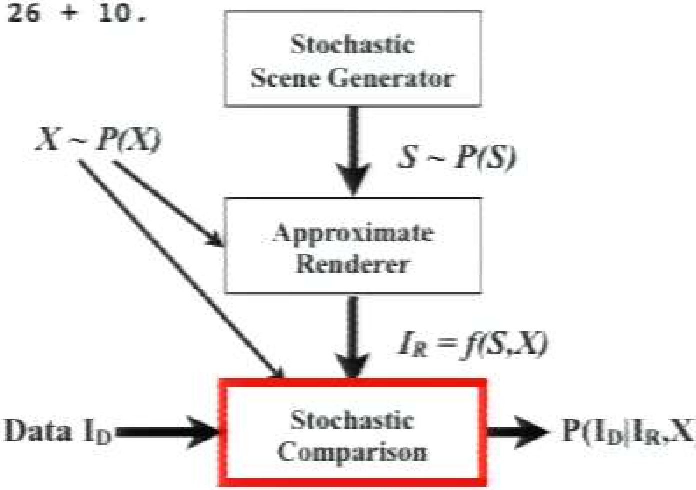
Probabilistic code in Venture, (Mansinghka, Selsam and Perov, in prep)
a new Turing-complete platform descended from Church (Goodman*, Mansinghka*, Roy et al., 2008)

GPGP Illustration:
Convergence issues without control variables



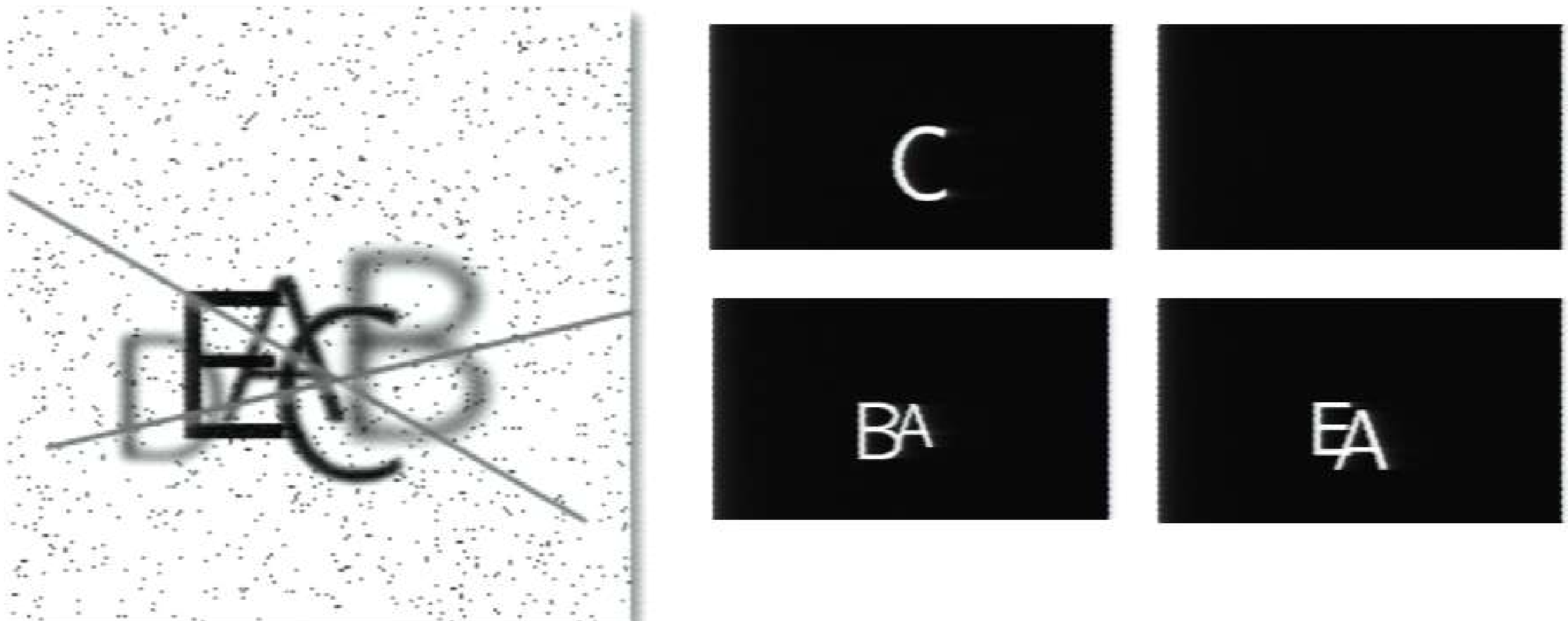
GPGP Illustration: Reading Obscured Text

```
ASSUME is_present (mem (lambda (id) (bernoulli 0.5)))
ASSUME pos_x (mem (lambda (id) (uniform_discrete 0 200)))
ASSUME pos_y (mem (lambda (id) (uniform_discrete 0 200)))
ASSUME size_x (mem (lambda (id) (uniform_discrete 0 100)))
ASSUME size_y (mem (lambda (id) (uniform_discrete 0 100)))
ASSUME rotation (mem (lambda (id) (uniform_continuous -20.0 20.0)))
ASSUME glyph (mem (lambda (id) (uniform_discrete 0 35))) // 26 + 10.

ASSUME blur (mem (lambda (id) (* 7 (beta 1 2))))
ASSUME global_blur (* 7 (beta 1 2))
ASSUME data_blur (* 7 (beta 1 2))
ASSUME epsilon (gamma 1 1)

ASSUME image (render_surfaces max-num-glyphs global_blur
(pos_x 1) (pos_y 1) (glyph 1) (size_x 1) (size_y 1)
(rotation 1) (blur 1)  (is_present 1)
(pos_x 2) (pos_y 2) (glyph 2) (size_x 2) (size_y 2)
(rotation 2) (blur 2) (is_present 2)
... (is_present 10))

ASSUME data (load_image "captcha_1.png" data_blur)
OBSERVE (incorporate_stochastic_likelihood data image epsilon) True
```

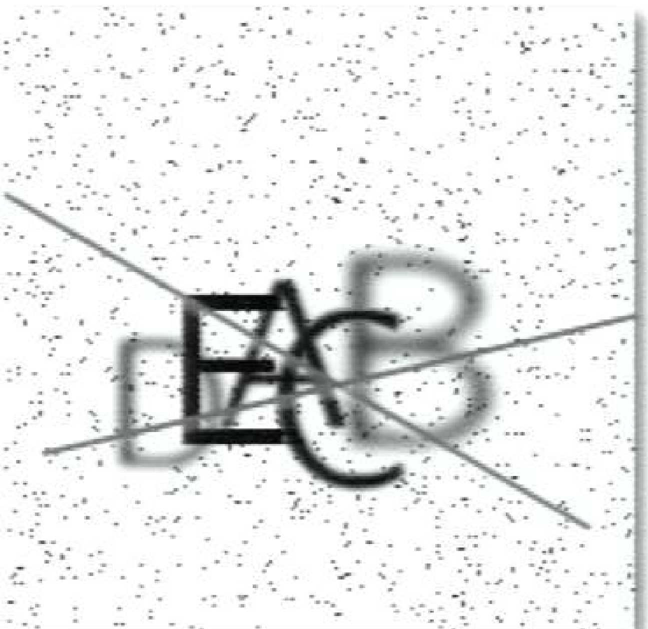


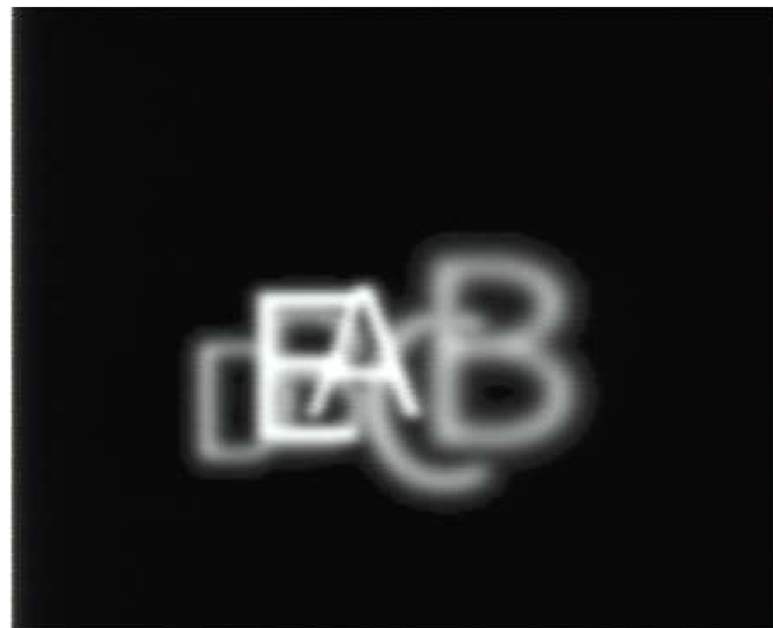
Probabilistic code in Venture, (Mansinghka, Selsam and Perov, in prep)
a new Turing-complete platform descended from Church (Goodman*, Mansinghka*, Roy et al., 2008)

GPGP Illustration:
Convergence issues without control variables

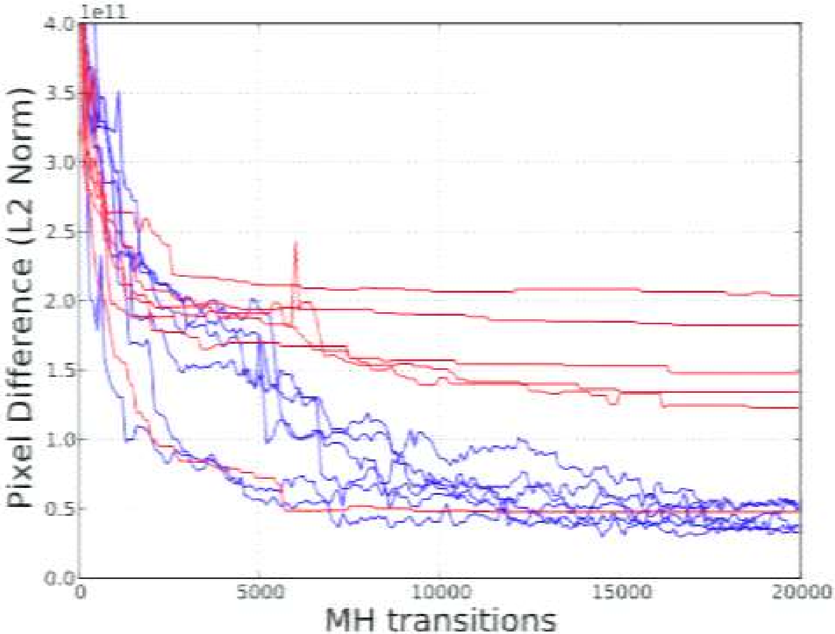


GPGP Illustration:
Improved convergence via Bayesian relaxations





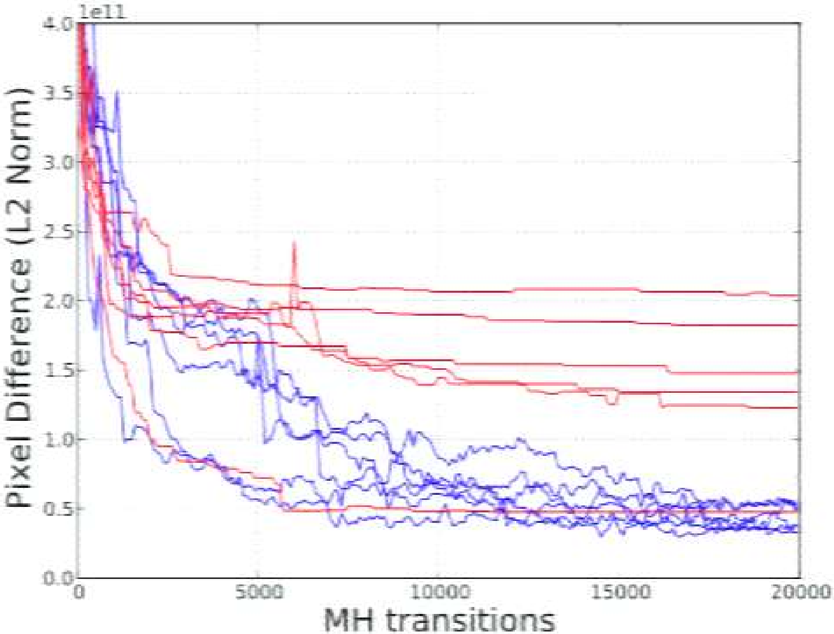
GPGP Illustration:
Improved convergence via Bayesian relaxations

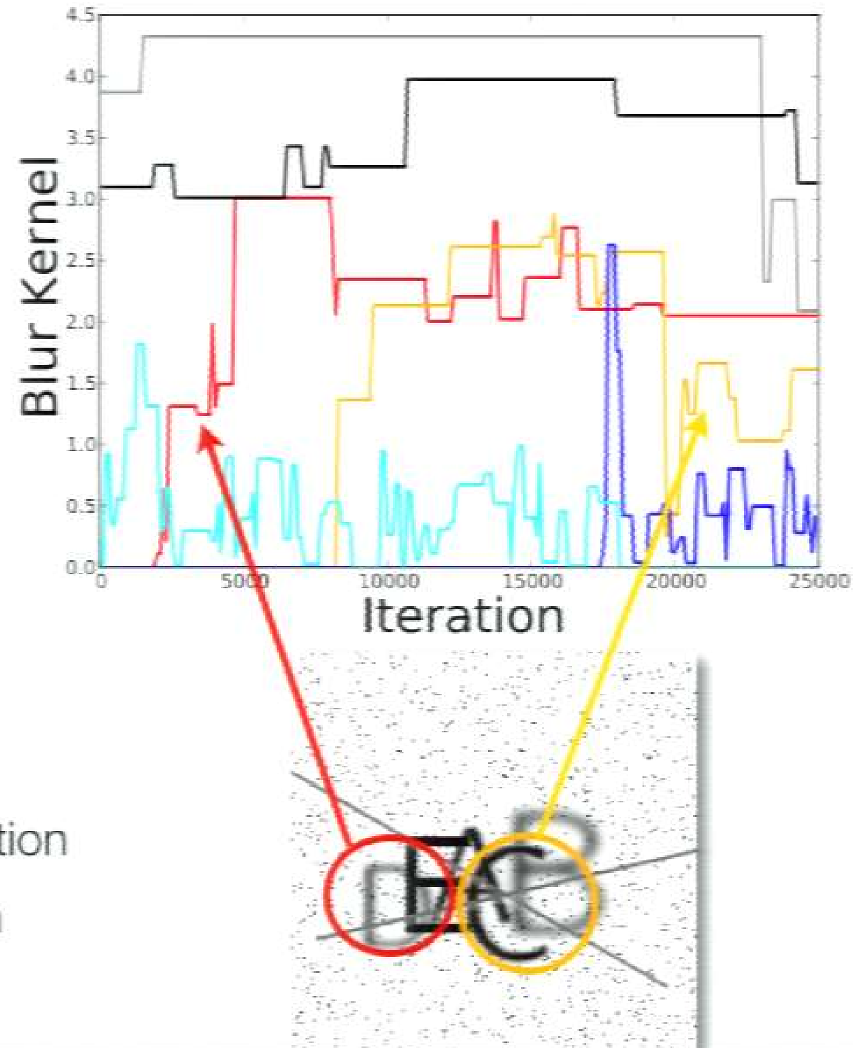


Without Bayesian relaxation

With Bayesian relaxation

GPGP Illustration:
Improved convergence via Bayesian relaxations

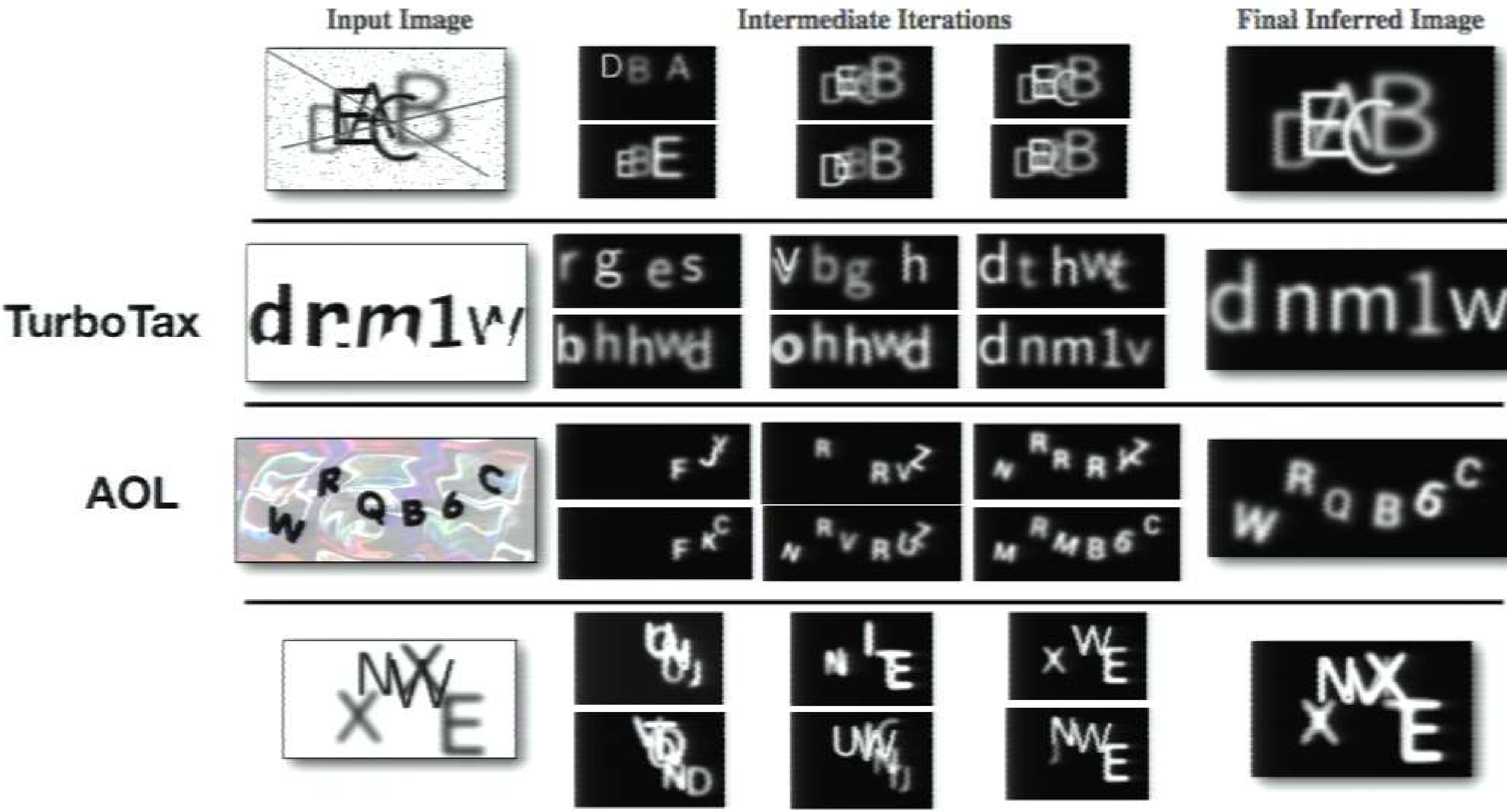




Without Bayesian relaxation

With Bayesian relaxation

GPGP Illustration: Empirical Results

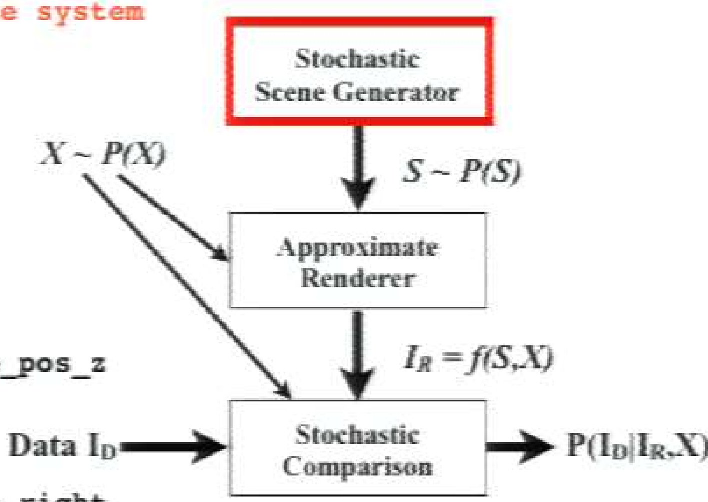


GPGP in 3D: Finding Roads

Scene from KITTI Vision Benchmark Suite:

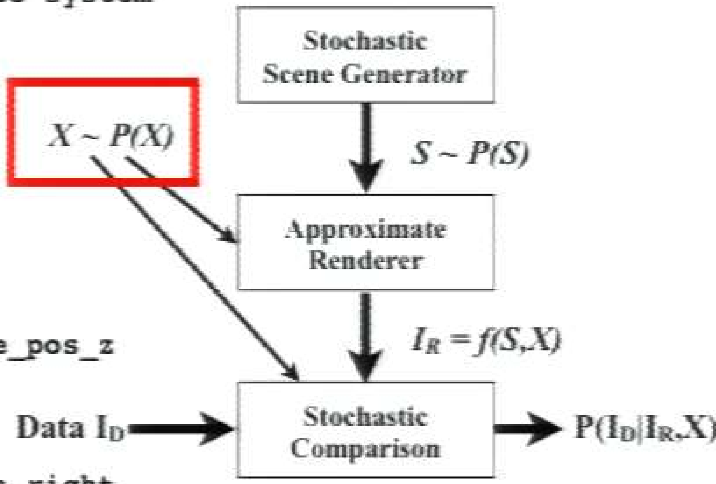
GPGP in 3D: The probabilistic code

```
ASSUME road_width (uniform_discrete 5 8) //arbitrary units
ASSUME road_height (uniform_discrete 70 150)
ASSUME lane_pos_x (uniform_continuous -1.0 1.0) //uncentered renderer
ASSUME lane_pos_y (uniform_continuous -5.0 0.0) //coordinate system
ASSUME lane_pos_z (uniform_continuous 1.0 3.5)
ASSUME lane_size (uniform_continuous 0.10 0.35)

ASSUME eps (gamma 1 1)
ASSUME theta_left (list 0.13 ... 0.03)
ASSUME theta_right (list 0.03 ... 0.02)
ASSUME theta_road (list 0.05 ... 0.07)
ASSUME theta_lane (list 0.01 ... 0.21)

ASSUME surfaces (render_surfaces lane_pos_x lane_pos_y lane_pos_z
  road_width road_height lane_size)

ASSUME data (load_image "frame201.png")
OBSERVE (incorporate_stochastic_likelihood theta_left theta_right
  theta_road theta_lane data surfaces eps) True
```



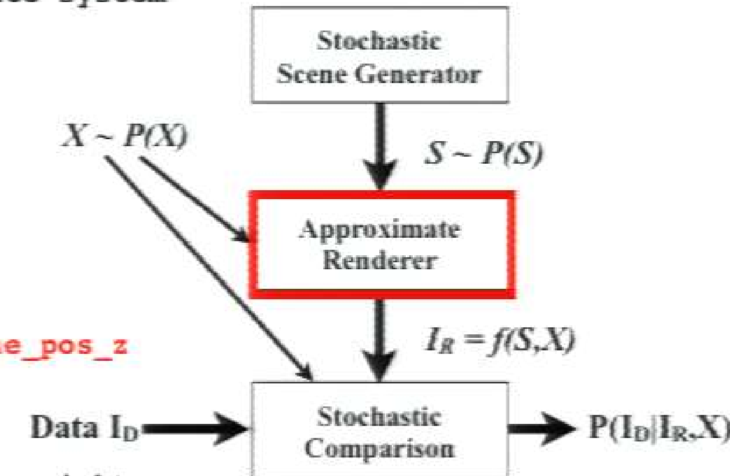
GPGP in 3D: The probabilistic code

```
ASSUME road_width (uniform_discrete 5 8) //arbitrary units
ASSUME road_height (uniform_discrete 70 150)
ASSUME lane_pos_x (uniform_continuous -1.0 1.0) //uncentered renderer
ASSUME lane_pos_y (uniform_continuous -5.0 0.0) //coordinate system
ASSUME lane_pos_z (uniform_continuous 1.0 3.5)
ASSUME lane_size (uniform_continuous 0.10 0.35)

ASSUME eps (gamma 1 1)
ASSUME theta_left (list 0.13 ... 0.03)
ASSUME theta_right (list 0.03 ... 0.02)
ASSUME theta_road (list 0.05 ... 0.07)
ASSUME theta_lane (list 0.01 ... 0.21)

ASSUME surfaces (render_surfaces lane_pos_x lane_pos_y lane_pos_z
  road_width road_height lane_size)

ASSUME data (load_image "frame201.png")
OBSERVE (incorporate_stochastic_likelihood theta_left theta_right
  theta_road theta_lane data surfaces eps) True
```



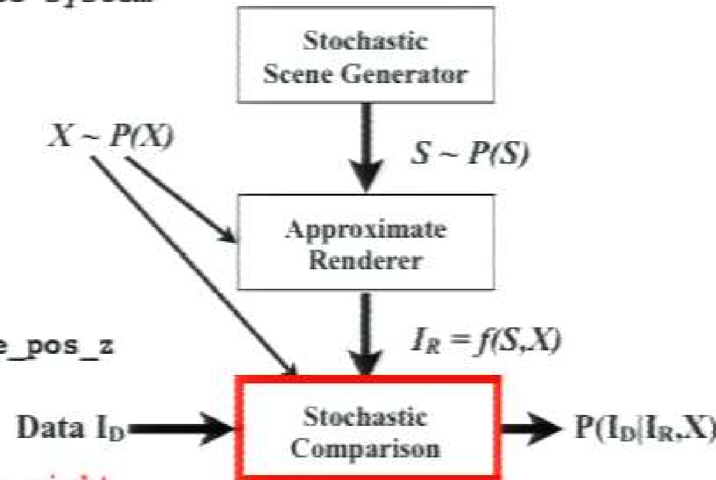
GPGP in 3D: The probabilistic code

```
ASSUME road_width (uniform_discrete 5 8) //arbitrary units
ASSUME road_height (uniform_discrete 70 150)
ASSUME lane_pos_x (uniform_continuous -1.0 1.0) //uncentered renderer
ASSUME lane_pos_y (uniform_continuous -5.0 0.0) //coordinate system
ASSUME lane_pos_z (uniform_continuous 1.0 3.5)
ASSUME lane_size (uniform_continuous 0.10 0.35)

ASSUME eps (gamma 1 1)
ASSUME theta_left (list 0.13 ... 0.03)
ASSUME theta_right (list 0.03 ... 0.02)
ASSUME theta_road (list 0.05 ... 0.07)
ASSUME theta_lane (list 0.01 ... 0.21)

ASSUME surfaces (render_surfaces lane_pos_x lane_pos_y lane_pos_z
  road_width road_height lane_size)

ASSUME data (load_image "frame201.png")
OBSERVE (incorporate_stochastic_likelihood theta_left theta_right
  theta_road theta_lane data surfaces eps) True
```



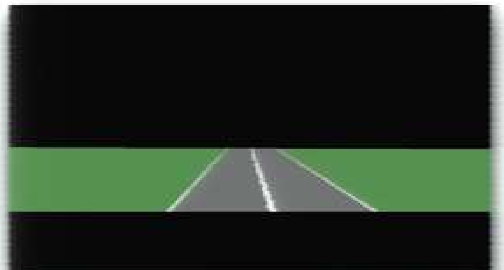
GPGP in 3D: The probabilistic code

```
ASSUME road_width (uniform_discrete 5 8) //arbitrary units
ASSUME road_height (uniform_discrete 70 150)
ASSUME lane_pos_x (uniform_continuous -1.0 1.0) //uncentered renderer
ASSUME lane_pos_y (uniform_continuous -5.0 0.0) //coordinate system
ASSUME lane_pos_z (uniform_continuous 1.0 3.5)
ASSUME lane_size (uniform_continuous 0.10 0.35)

ASSUME eps (gamma 1 1)
ASSUME theta_left (list 0.13 ... 0.03)
ASSUME theta_right (list 0.03 ... 0.02)
ASSUME theta_road (list 0.05 ... 0.07)
ASSUME theta_lane (list 0.01 ... 0.21)

ASSUME surfaces (render_surfaces lane_pos_x lane_pos_y lane_pos_z
  road_width road_height lane_size)

ASSUME data (load_image "frame201.png")
OBSERVE (incorporate_stochastic_likelihood theta_left theta_right
  theta_road theta_lane data surfaces eps) True
```



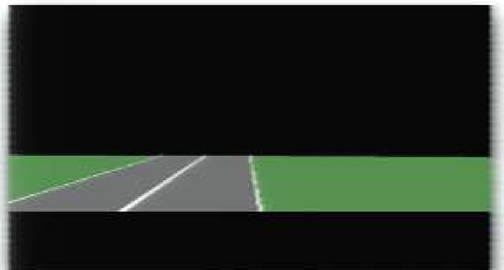
GPGP in 3D: The generative model

3D Scene Prior



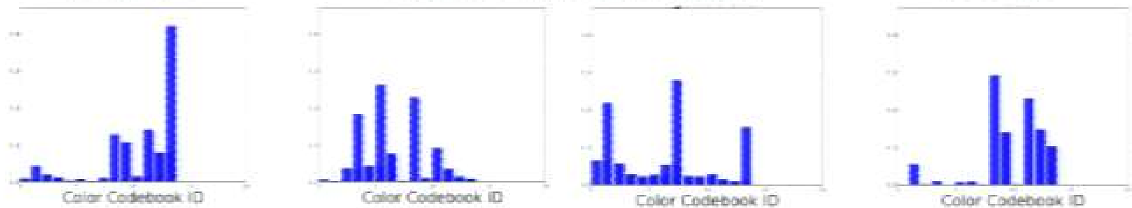
GPGP in 3D: The generative model

3D Scene Prior



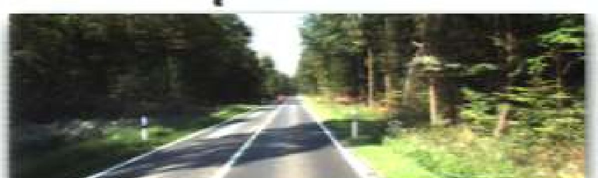
Histogram Appearance Models

Lanes | Offroad (left, right) | Road



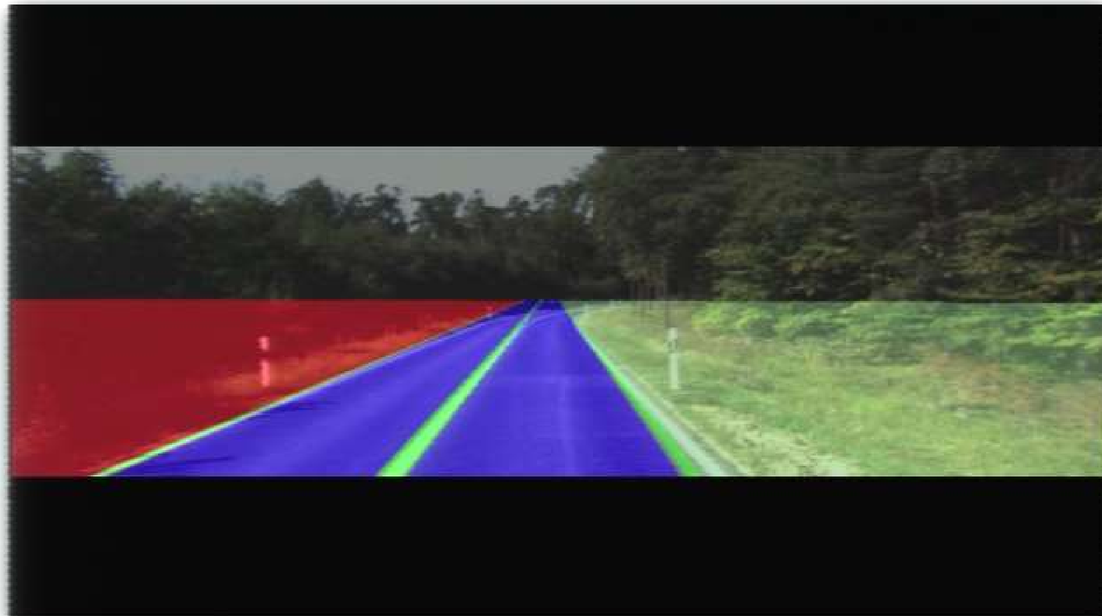
$$P(I_D|I_R,\epsilon) = \prod_{r\in R} \prod_{x,y \text{ s.t. } I_R=r} \frac{\theta_r^{I_{D(x,y)}} + \epsilon}{Z_r}$$

Input Data



Quantized Image

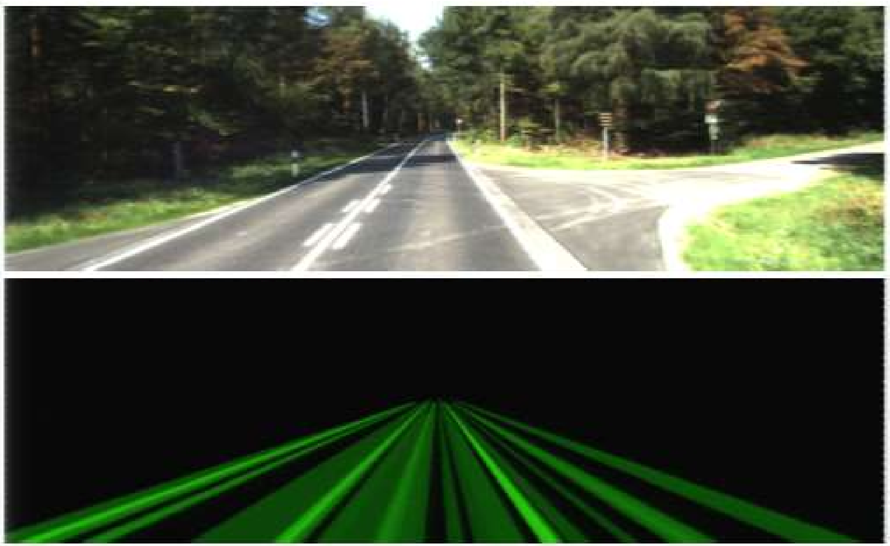
GPGP in 3D: Empirical Results



Method	Accuracy
Aly et al [1]	68.31%
GPGP (Best Single Appearance)	64.56%
GPGP (Maximum Likelihood over Multiple Appearances)	74.60%

GPGP in 3D: Posterior Uncertainty

Assumptions violated: broad posterior

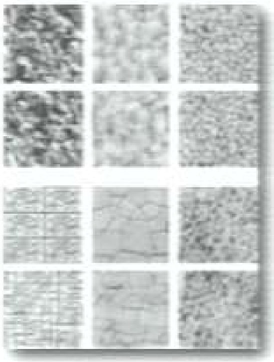


Assumptions satisfied: narrower posterior

Scaling up by
Integrating Knowledge Engineering and Learning

Scaling up by
Integrating Knowledge Engineering and Learning

- Learn parameterized generative models for appearance and shape:



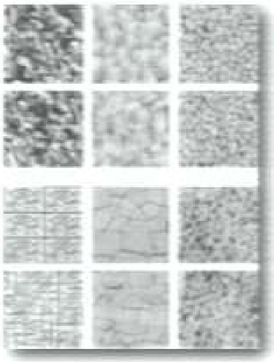
(Portilla & Simoncelli, 1999)

(Tang & Salakhutdinov, NIPS 2013)

Shape programs written in GML: (Havemann, 2005)

Scaling up by Integrating Knowledge Engineering and Learning

- Learn parameterized generative models for appearance and shape:



(Portilla & Simoncelli, 1999)

(b) Generated Horses

(Tang & Salakhutdinov, NIPS 2013)

Shape programs written in GML: (Havemann, 2005)

- Learn structured bottom-up inference programs automatically, from forward executions of the generative probabilistic graphics program:

Conclusion

- **Direct formulations of approximately Bayesian inverse graphics are possible, given:**

  1. Generative models written as probabilistic graphics programs in Church/Venture
  2. Automatic, general-purpose samplers for inference; no custom inference code needed
  3. Approximate comparison of rendering and image data: a variation on ABC
  4. Bayesian relaxations, to adaptively smooth the energy landscape

- **Links:**

  GPGP: http://probcomp.csail.mit.edu/gpgp
  Venture (alpha 0.1.1): http://probcomp.csail.mit.edu/venture
  Probabilistic Programming: http://probabilistic-programming.org
  DARPA PPAML: http://ppaml.galois.com

- **Acknowledgements:** Keith Bonawitz, Eric Jonas, Bill Freeman, Seth Teller and Max Siegel

NIPS Thanks Its Sponsors

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

UNITED STATES AIR FORCE

amazon.com

Microsoft Research

Google

facebook

SKYTREE

THE MACHINE LEARNING COMPANY

TWO SIGMA

United Technologies Research Center

D E Shaw & Co

xerox

YAHOO! LABS

IBM Research

DRW TRADING GROUP

TOYOTA

millionshort

criteo

PDT PARTNERS

Springer

Machine Learning Journal

Disney Research