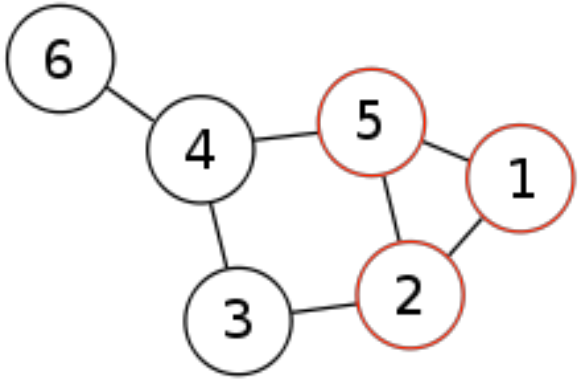
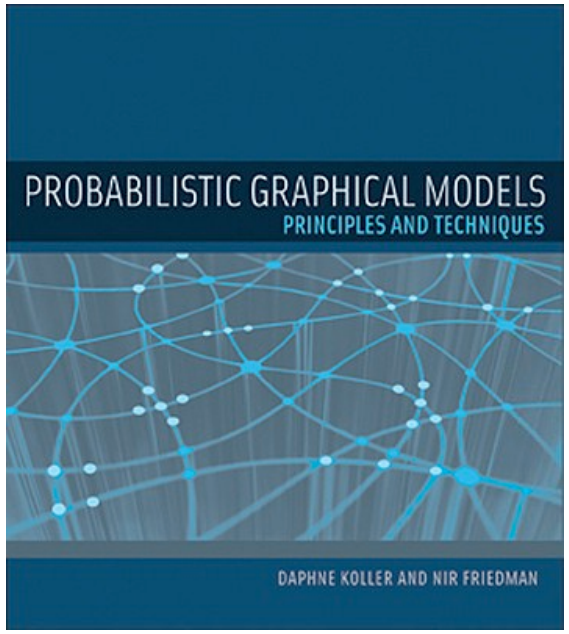


Learning in Undirected Graphical Models

Max Welling

UC Irvine

Overview



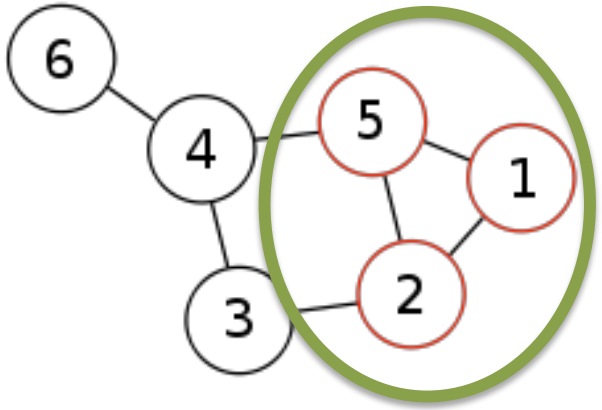
- MRF Representation.
- Varieties of MRF:
  - Fully observed, CRF, Hidden Units.
- Relation to “Maximum Entropy”.
- Gradients, Hessian, Convexity.
- Gradient descend with exact gradients (low tree-width):
  - MRF, CRF, Hidden Units.
- ML-BP and Pseudo-Moment Matching.
- Alternative Objectives:
  - PL, Contrastive Free Energies, Contrastive Divergence,
- Experiments
- More methods: MCMC-MLE, Composite Likelihood, Persistent CD.
- Herding
- Max Margin Markov Networks
- Structure Learning through L1-regularization
- Bayesian Learning:
  - Langevin-CD, Laplace Approximation-BP (EP, MCMC).
- Experiments
- Conclusions

MRF Representation: Gibbs Distribution

Assume a discrete sampling space X. Call X_k the group of variables in cluster "k".

$$P(X) = \frac{1}{Z[\Psi]} \prod_k \Psi_k(X_k), \quad \Psi_k(X_k) > 0$$

$$Z[\Psi] = \sum_{X^*} \prod_k \Psi_k(X_k^*)$$



Partition Function

$$P(X) = \frac{1}{Z[\theta]} \exp\left(\sum_k \theta_k f_k(X_k) \right)$$

$$Z[\theta] = \sum_{X^*} \exp\left(\sum_k \theta_k f_k(X_k^*) \right)$$

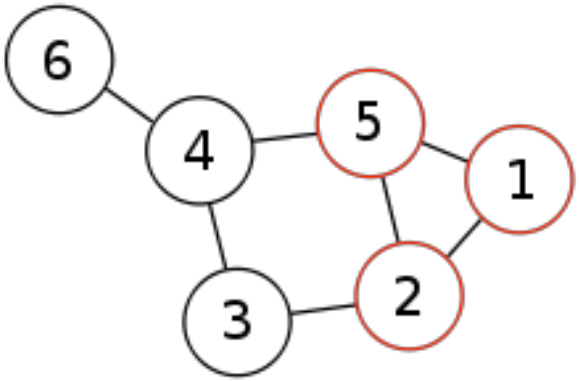
Conditional Independencies

- Consider a *positive* Gibbs distribution that factorizes as: $P(X) = \frac{1}{Z[\Psi]} \prod_k \Psi_k(X_k) > 0$

- Consider a graphical representation G where all variables that reside inside a factor are connected by an edge (a clique).

- Consider the following conditional independence relationships (CIRs) for G:
  $X \perp Y \mid Z$ if every path between X and Y is blocked by nodes in Z.

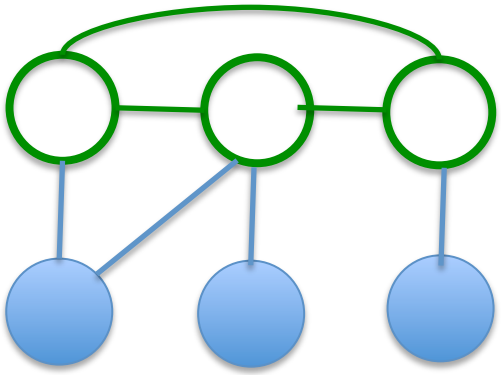
- Then the CIRs for the graph G "correspond" to the CIRs in P.
  - -any P that factorizes over G satisfies these CIRs.
  - -if X and Y is not blocked by Z than there exists a P that factorizes over G in which they are dependent given Z.



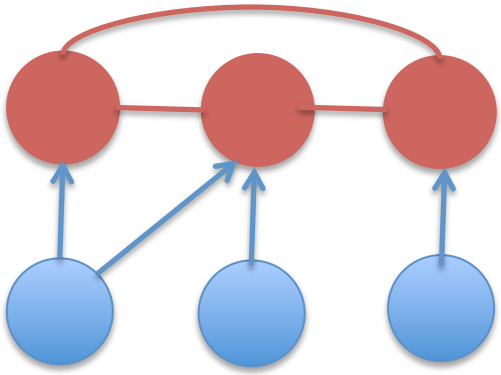
Example CIR: $X_6 \perp X_3 \mid X_4$

Note: P could have a single factor f(1,2,5) or 3 factors f(1,2), f(1,5), f(2,5).

Hidden Units & Conditional Models

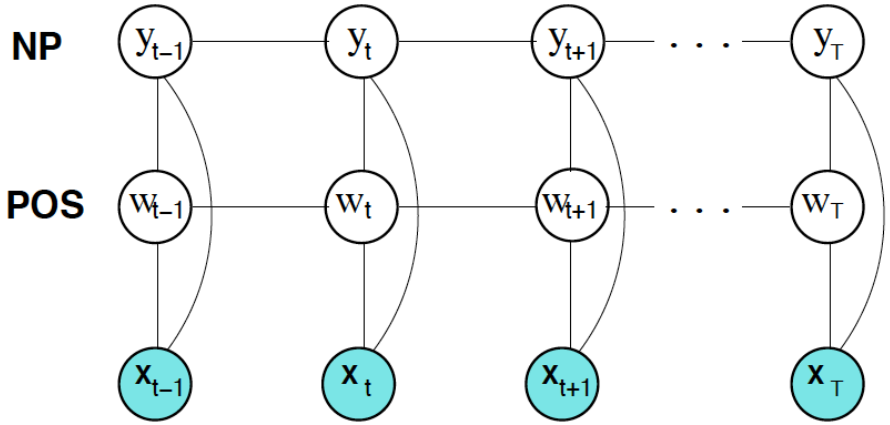


$$P(X) = \frac{Z(\theta, X)}{Z(\theta)} = \frac{\sum_{Z} \exp\left(\sum_{k} \theta_k f_k(X_k, Z_k)\right)}{\sum_{X^*, Z^*} \exp\left(\sum_{k} \theta_k f_k(X_k^*, Z_k^*)\right)}$$



$$P(Y \mid X) = \frac{\exp\left(\sum_{k} \theta_k f_k(Y_k, X_k)\right)}{\sum_{Y^*} \exp\left(\sum_{k} \theta_k f_k(Y_k^*, X_k)\right)}$$

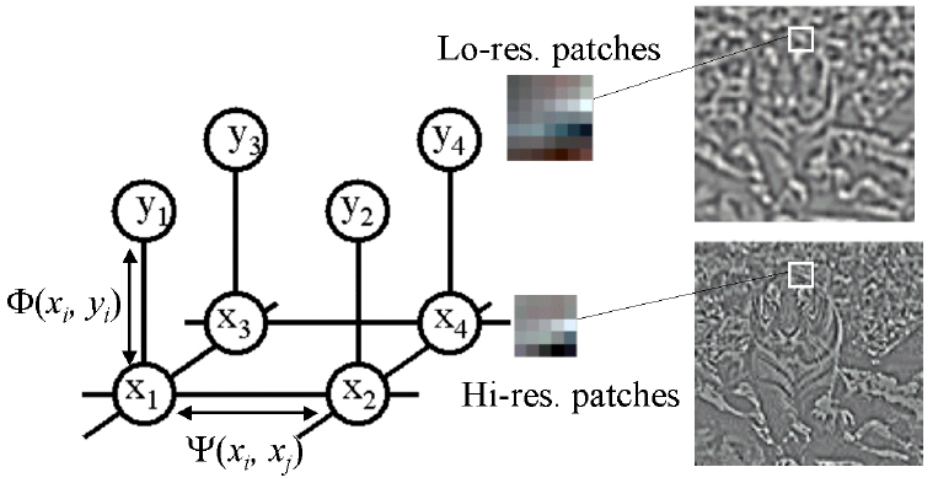
Examples



Graphical representation of factorial CRF for the joint noun phrase chunking and part of-speech tagging problem, where the chain y represents the NP labels, and the chain w represents the POS labels

Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. Charles Sutton, Andrew McCallum, Khashayar Rohanimanesh. Journal of Machine Learning Research 8. 2007

Markov network model for super-resolution problem



Example-based super-resolution
William T. Freeman, Thouis R. Jones, and Egon C. Pasztor.

Maximum Entropy

$$\underset{P}{Max} \; -\sum_X P(X)\log P(X)$$ Jaynes

$$subject \;\; to: \;\; E_P[f_k] = E_{\hat{P}}[f_k], \;\; k = 1,..,K$$

$$\hat{P}(X) = \frac{1}{n}\sum_i \delta(X - x_i)$$ (empirical distribution)

Idea: We want to find a probability distribution P with moments $E_P[f_k] = E_{\hat{P}}[f_k]$.
However, there are many such P's.
Find the one with the least "additional assumptions" = maximum entropy.

Maximum entropy problem has the same solution as its dual: maximum likelihood problem.

$$\underset{\theta}{Max} \; L(Data,\theta) = \underset{\theta}{Max} \; \sum_k \theta_k E_{\hat{P}}[f_k(X_k)] - \log Z(\theta) = \underset{\theta}{Max} \; \frac{1}{n}\sum_{i=1}^{n} \log P(x_i)$$

With: $P(X) = \frac{1}{Z(\theta)}\exp(\sum_k \theta_k f_k(X_k))$;

H(P) is the dual of log(Z), parameters play role of Lagrange multipliers.

Gradients & Hessian of L

The log-likelihood of the data for the model P is given as:

$$L(Data,\theta) = \sum_{i=1}^{n}\sum_{k}\theta_k f_k(x_{ik}) - n\log Z(\theta)$$
$$= n(\sum_{k}\theta_k E_{\hat{P}}[f_k] - \log Z(\theta))$$

To optimize it we would need the first derivative:

$$\nabla_{\theta_k} L(Data,\theta) = n(E_{\hat{P}}[f_k] - E_{P_\theta}[f_k])$$

If we divide by "n" we see that the direction of steepest ascent is the difference between two averages: the average of the data and the average of the model.

Hessian is given by:

$$H_{kl} = \nabla_{\theta_k}\nabla_{\theta_l} L(Data,\theta) = -nCov_{P_\theta}[f_k, f_l]$$

Since the Covariance is positive definite we conclude that *L is concave!*
(this result only holds if there are no hidden variables).

Gradient Descent MRF

Gradient descent:

$$\theta_k \leftarrow \theta_k + \frac{\eta}{n}\nabla_{\theta_k} L(Data,\theta)$$
$$= \theta_k + \eta(E_{\hat{P}}[f_k] - E_{P_\theta}[f_k])$$

Looks simple, however: $E_{P_\theta}[f_k]$ is intractable in MRF with high treewidth.

We could approximate the gradient using approximate inference techniques.
For instance we could use:

1) Loopy Belief Propagation. This is "fast" but may not converge as parameters get stronger. Also, the estimate of the gradients is biased at every iteration so we may end up far away from the true optimum.
2) MCMC. This is slow because we need to run a MCMC chain to convergence for every update. Also, if we don't collect enough samples to average over we will have high variance in the estimates of the gradients.
3) Can we trade off bias and variance?

GD for CRF

$$\nabla_{\theta_k} L = \sum_{i=1}^{n} \left(f_k(y_{ik}, x_{ik}) - E_{P_\theta(Y|x_i)}[f_k(Y_k, x_{ki})] \right)$$

$$\nabla_{\theta_k} \nabla_{\theta_l} L = -\sum_{i=1}^{n} Cov_{P_\theta(Y|x_i)}[f_k(Y_k, x_{ki}) f_l(Y_l, x_{li})]$$

- Even for the gradients we need to perform inference separately for every data-case "i".
- However, the inference is only over the labels Y, not the feature space X.
- Conditioning on evidence often makes the distribution peak at a small subset of states, i.e. there are fewer modes in P(Y|x).
- To learn good conditional models for Y|x we often need more data than for generative models over Y,X. Our assumptions on P(X) regularize learning.
  This reduces variance (and thus over-fitting) but may introduce bias.

GD for PO-MRF

$$\nabla_{\theta_k} L = \sum_{i=1}^{n} \left(E_{P_\theta(Z_k|x_i)}[f_k(Z_k, x_{ik})] \right) - nE_{P_\theta(Z,X)}[f_k(Z_k, X_k)]$$

$$\nabla_{\theta_k} \nabla_{\theta_l} L = -\left(nCov_{P_\theta(Z,X)}[f_k(Z_k, X_k) f_l(Z_l, X_l)] - \sum_{i=1}^{n} Cov_{P_\theta(Z|x_i)}[f_k(Z_k, x_{ik}) f_l(Z_l, x_{il})] \right)$$

- Evaluating gradient involves inference over Z|x for every data-case PLUS one more time on Z,X jointly without any evidence clamped.

- Hessian is now no longer a covariance but a difference of covariances which can easily get *negative* definite => L is no longer concave as a function of its parameters!

Pseudo-Moment Matching (1)

- Simplest case: fully observed MRF.
- Even in this case we need to estimate the intractable quantity $E_{P_\theta}[f_k]$ if we want to do gradient descent.

$$\theta_k \leftarrow \theta_k + \frac{\eta}{n}\nabla_{\theta_k} L = \theta_k + \eta(E_{\hat{P}}[f_k] - E_{P_\theta}[f_k])$$

- One idea is to approximate this expectation by running loopy belief propagation.
- Wainwright argues that this is reasonable if both learning and testing use the same approximate inference algorithm.
- Unfortunately, as the parameters grow LBP is less likely to converge.
- For the fully observed case, one can actually compute an analytical expression for the final answer, called *pseudo-moment matching*. [Wainwright,Jaakkola,Willsky '03]

Pseudo-Moment Matching (2)

- Consider the following special case:
  - MRF is fully observed.
  - Features are state indicators for single variables and pairs of variables:
    $\{f_k(X_k)\} = \{I[X_i = s], I[X_i = s \wedge X_j = t]\}$(one for every node and edge in G)

  - Then the Pseudo-Moment Matching solution is:

$$\theta_{is} = \log \hat{P}(X_i = s), \qquad \theta_{ij,st} = \log \frac{\hat{P}(X_i = s, X_j = t)}{\hat{P}(X_i = s)\hat{P}(X_j = t)}$$

  - We are now guaranteed that at one of LBP fixed points we have:

$$P_{LBP}(X_i) = \hat{P}(X_i), \quad P_{LBP}(X_i, X_j) = \hat{P}(X_i, X_j)$$

  - Generalization to higher order indicator features and GBP exist.

  - Generalization to hidden units not clear.

  - Solution only uses local information while we know that Z couples all parameters.

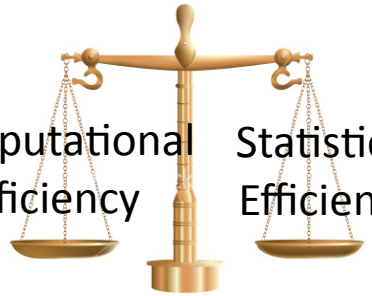
Pseudo-Likelihood (1)

Besag '75

- Instead of approximating the gradients we can also change the objective.
- In the PL we replace: $P(X_1,...,X_D) \rightarrow \prod_{k=1}^{D} P(X_k \mid X_{-k})$.
- We maximize: $PL = \sum_{i=1}^{n}\sum_{k=1}^{D} \log P(x_{ki} \mid x_{-k,i})$
- As $n \rightarrow \infty$, $\theta_{PL} \xrightarrow{P} \theta_{true}$ which we call "consistency".

- However, PL is less statistically efficient than ML.
  This means that we would repeat the estimation procedure with n data-cases many times, then the average will be correct (unbiased) but the variance will larger than for ML.

- Computationally, PL is much more efficient however because we only need to normalize conditional distributions over a single variable.



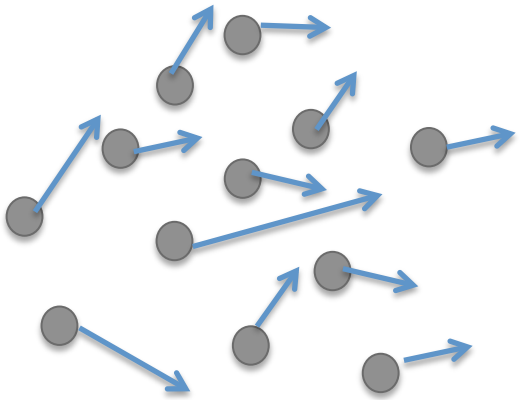
Pseudo-Likelihood (2)

$$\nabla_{\theta_k} PL = nE_{\hat{P}}[f_k] - \frac{1}{|X_k|} \sum_{j:X_j \in X_k} \sum_{i=1}^{n} E_{P(X_j | x_{-j,i})}[f_k(X_j, x_{[k \setminus j],i})]$$

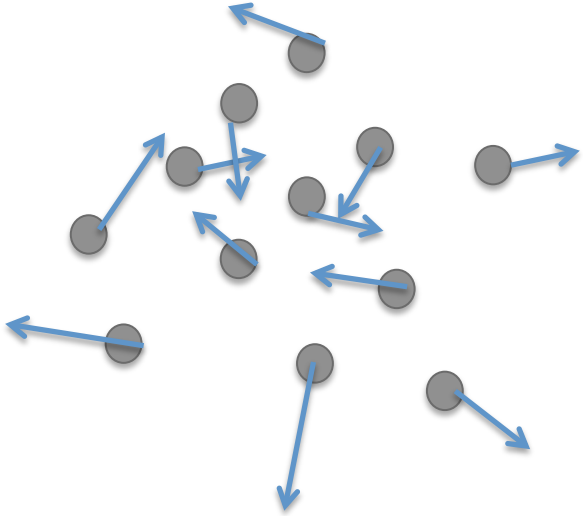
with:
$$P(X_j \mid X_{-j}) = \frac{\exp[\sum_k \theta_k f_k(X_k)]}{\sum_{X_j} \exp[\sum_k \theta_k f_k(X_j, X_{k \setminus j})]}$$

- Note that it is easy to compute these conditionals because the summation is over values of only one variable.
- Hessian is a little involved but negative definite, so the PL is a concave objective.
- PL doesn't work for partially observed MRFs. (Some parameters may diverge.)
- Intuition: Think of every data-case as a sample. For each of them, relax one variable and let it re-sample from $P(X_j \mid x_{-ji})$. If the new samples so obtained are different than the original samples, then you need to change the parameters. The model is not perfect yet.

Contrastive Free Energies



- We relax the data-samples using some procedure.
- If they systematically change the E[f], then the parameters are imperfect and need to be updated.
- For PL, we relax one variable at a time (one dimension) and sum these contributions.



- We can also relax by running a brief MCMC procedure starting at every data-case.
- Or we can relax a variational distribution Q on a mean field free energy or the Bethe free energy.

- Analogy (Hinton):
  1. I will tell you a story (data)
  2. You will explain it back to me (relax the data, mix)
  3. As soon as you make a mistake I will correct you and you will have to start over again (parameter update)

Contrastive Divergence (1)

Hinton '02

REPEAT:

1. Randomly select a mini-batch of data items of size n.

2. Run n MCMC chains for r steps starting at the data items in the mini-batch.

3. Compute feature expectations under the data distribution and the r-step samples: $E_{\hat{P}}[f_k], E_{\hat{Q}^r}[f_k]$

4. Update the parameters of the model according to*: $\theta_k \leftarrow \theta_k + \eta \, (E_{\hat{P}}[f_k] - E_{\hat{Q}^r}[f_k])$

5. Increase the number of steps r (this decreases bias).

- Because we initialize the MCMC chains at the data-points (and not randomly) and we run for only r steps we decrease *the variance* in the gradient estimate.

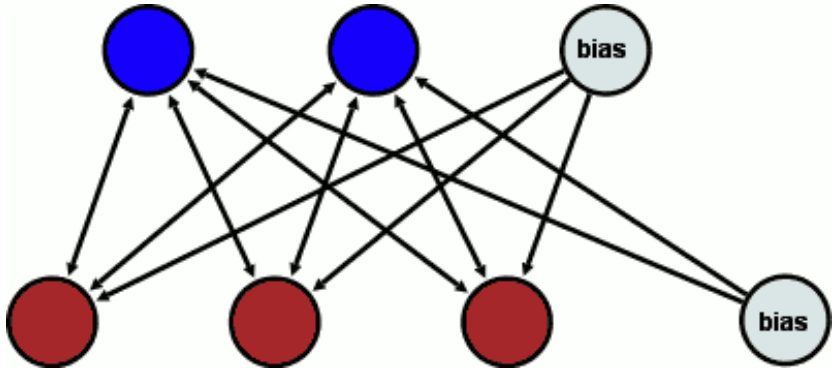
- However, because we sample for only r steps we increase *the bias.*

- As we increase r we trade less bias for more variance and computation.

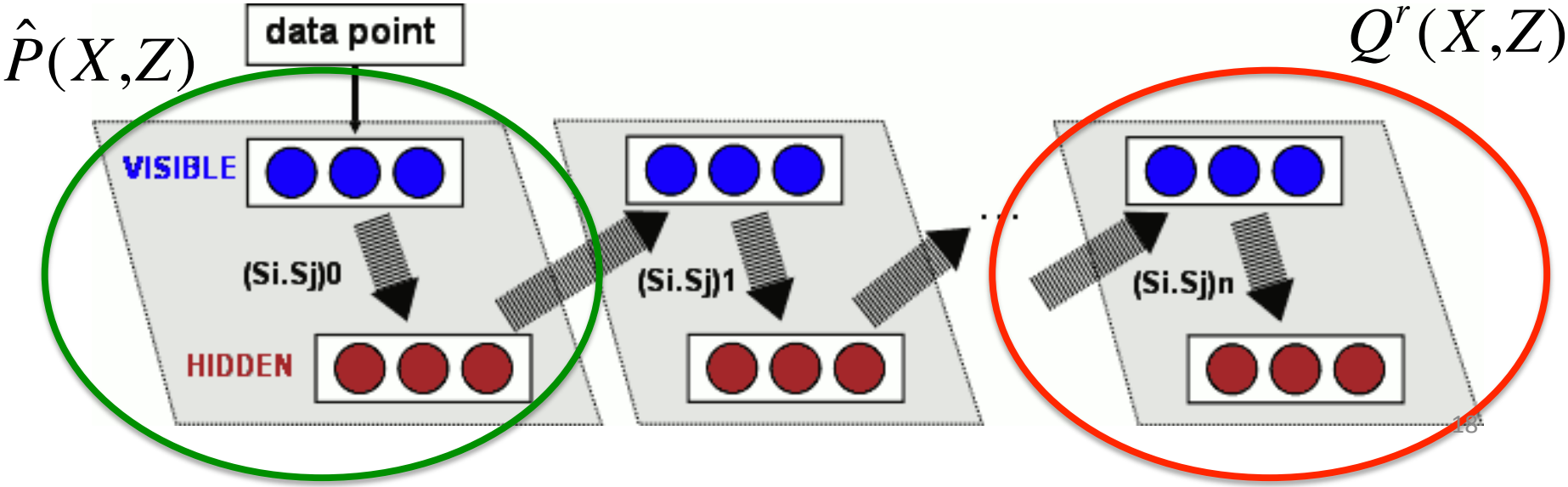
* This is *not* the gradient of the contrastive divergence: $KL[\hat{P} \| P_\theta] - KL[\hat{Q}^r \| P_\theta)$

Contrastive Divergence-RBM(2)

- Restricted Boltzmann Machine (RBM) has a layer of observed variables and a layer of unobserved variables , but no connections between them. All variables are *binary valued*.



- CD works well for this type of architecture because $P(Z \mid X) = \prod_j P(Z_j \mid X)$ & $P(X \mid Z) = \prod_i P(X_i \mid Z)$
- Update rule the same $\theta_k \leftarrow \theta_k + \eta \, (E_{\hat{P}}[f_k] - E_{\hat{Q}^r}[f_k])$



Contrastive Divergence -Example (3)

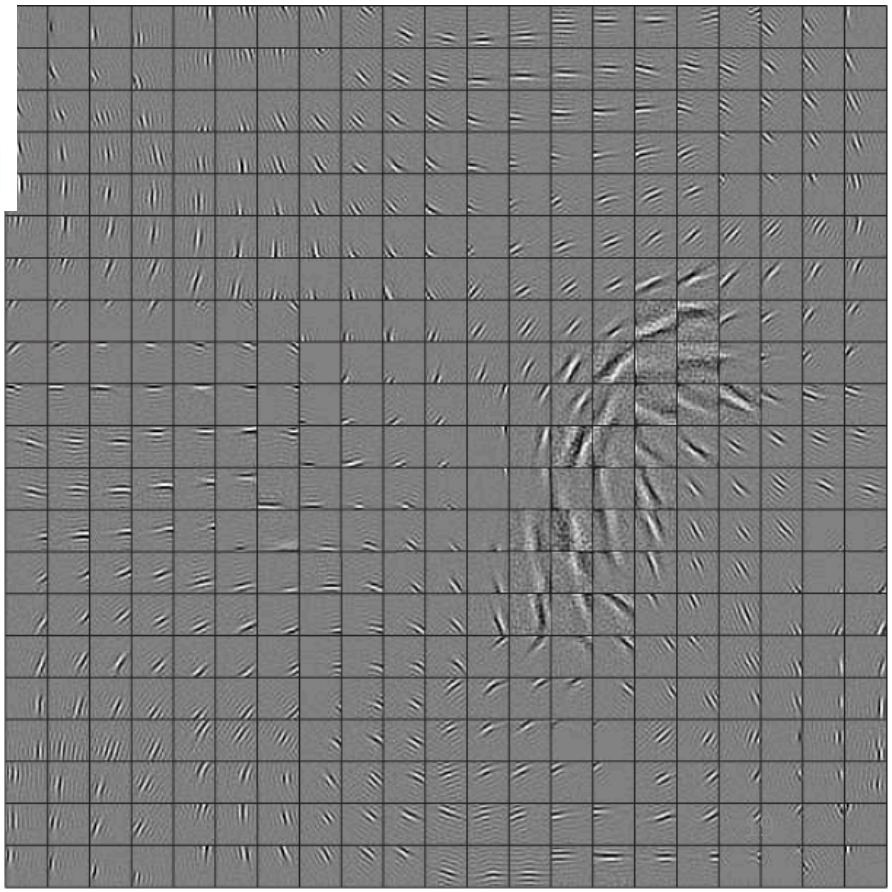
Osindero,et al '05

$$P(\mathbf{x},\mathbf{u}) \propto \exp\left[-\sum_{i=1}^{M}\left(u_i\left(1+\frac{1}{2}\sum_{j=1}^{K}W_{ij}(\mathbf{J}_j\mathbf{x})^2\right)+(1-\alpha_i)\log u_i\right)\right]$$

$$P(\mathbf{u}|\mathbf{x}) = \prod_{i=1}^{M}\mathcal{G}_{u_i}\left[\alpha_i\ ;\ 1+\frac{1}{2}\sum_{j=1}^{K}W_{ij}(\mathbf{J}_j\mathbf{x})^2\right]$$

$$P(\mathbf{x}|\mathbf{u}) = \mathcal{N}_{\mathbf{x}}\left[0\ ;\ (\mathbf{J}\mathbf{V}\mathbf{J}^T)^{-1}\right] \qquad \mathbf{V}=\mathbf{Diag}[\mathbf{W}^T\mathbf{u}]$$

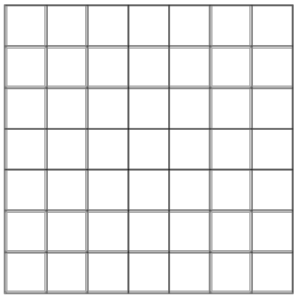
J



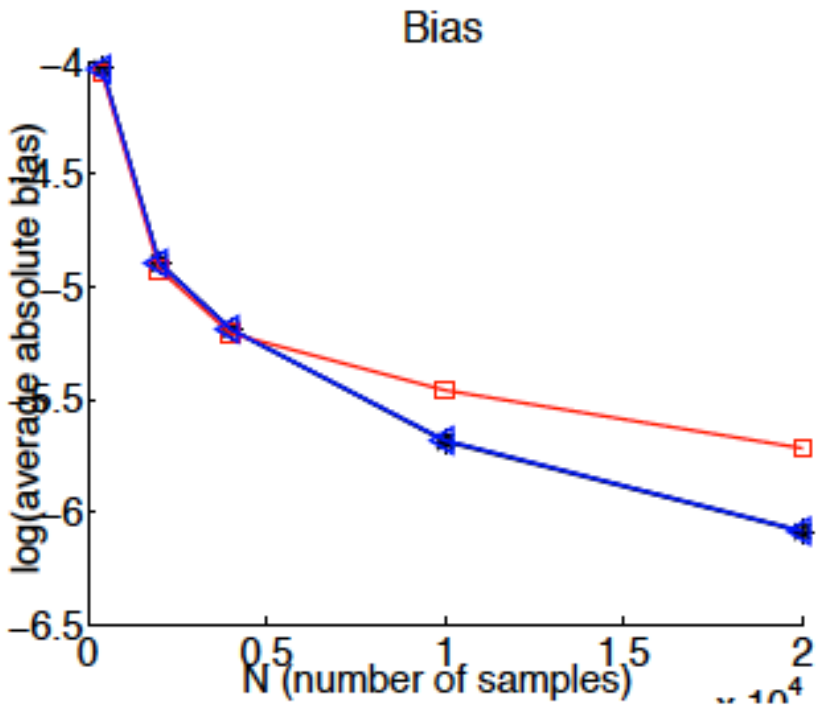
- Data: X: lowpass filtered greyscale images.
- U=Z are hidden variables that model the precision for the conditional Gaussian P(X|U)
- Variables s=Jx are organized on a 2-d grid, and W only connects to a small neighborhood of s-variables.
- x,u are continuous.

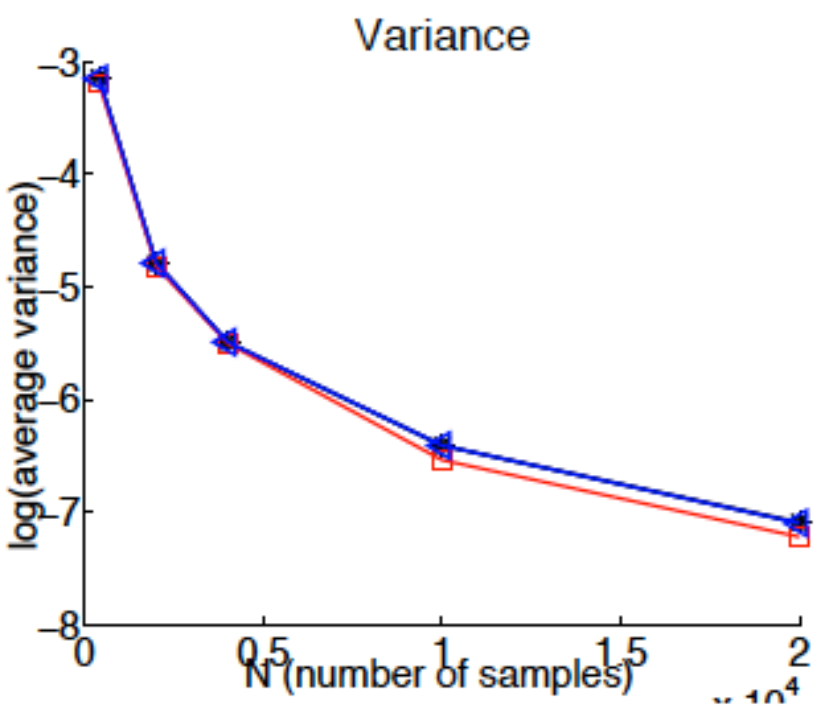
Some Experimental Results

Binary values (Boltzmann machine)
Fully observed
Square grid of size 7x10.
Parameters sampled from U[-0.1,0.1]
Vary nr of data-cases sampled from model



Parise, Welling '05

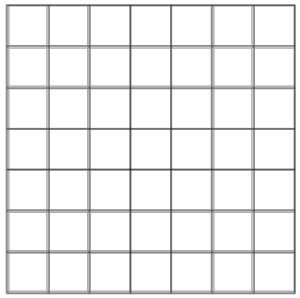




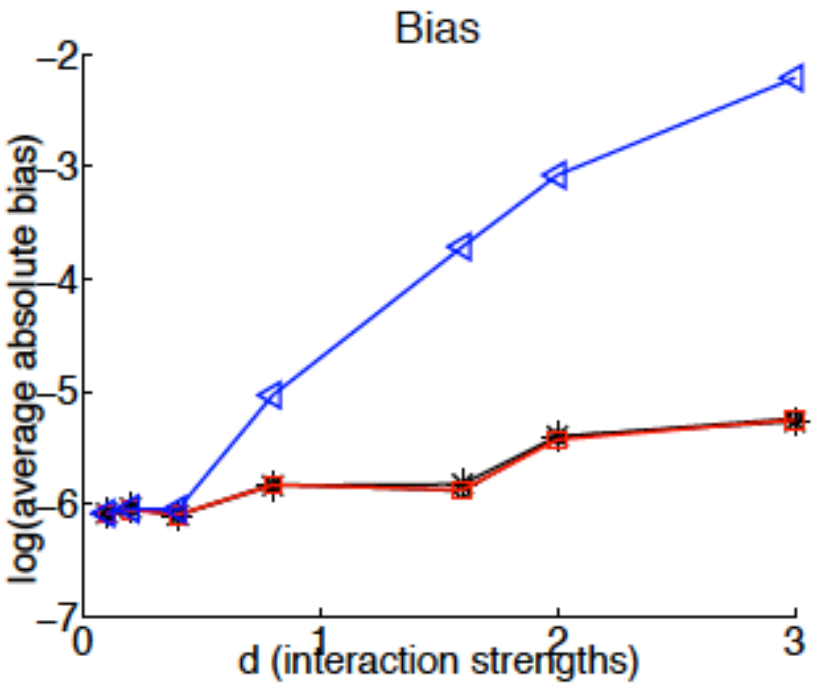
FO-grid-N Experiment (○,∗ , □, ◁ indicate ML-exact, PL, CD (K=5), PMM respectively)

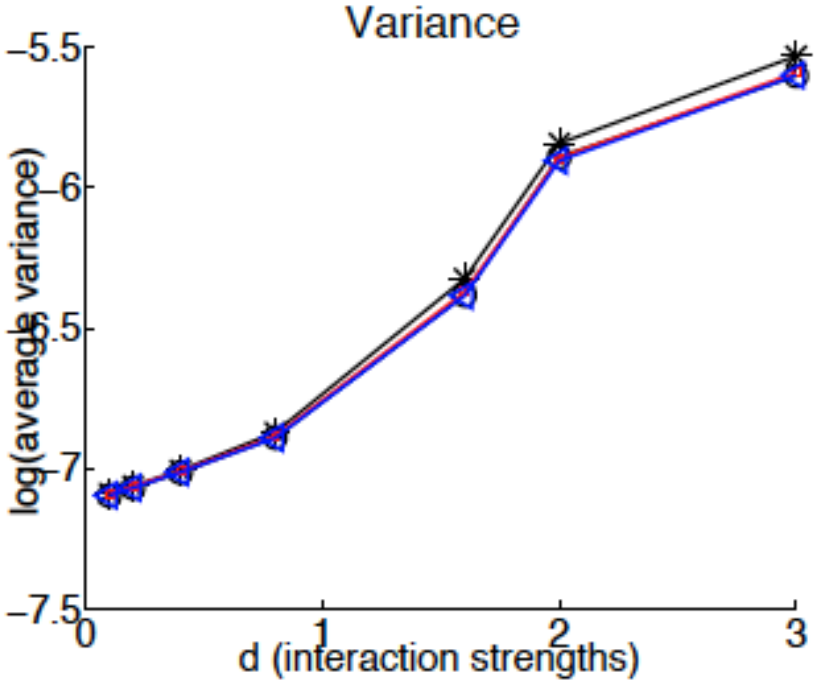
Some Experimental Results

Binary values (Boltzmann machine)
Fully observed
Square grid of size 7x10.
Parameters sampled from U[-d/2,d/2], vary d
N = 2000



Parise, Welling '05

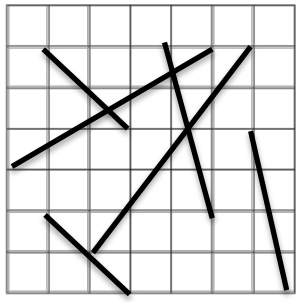




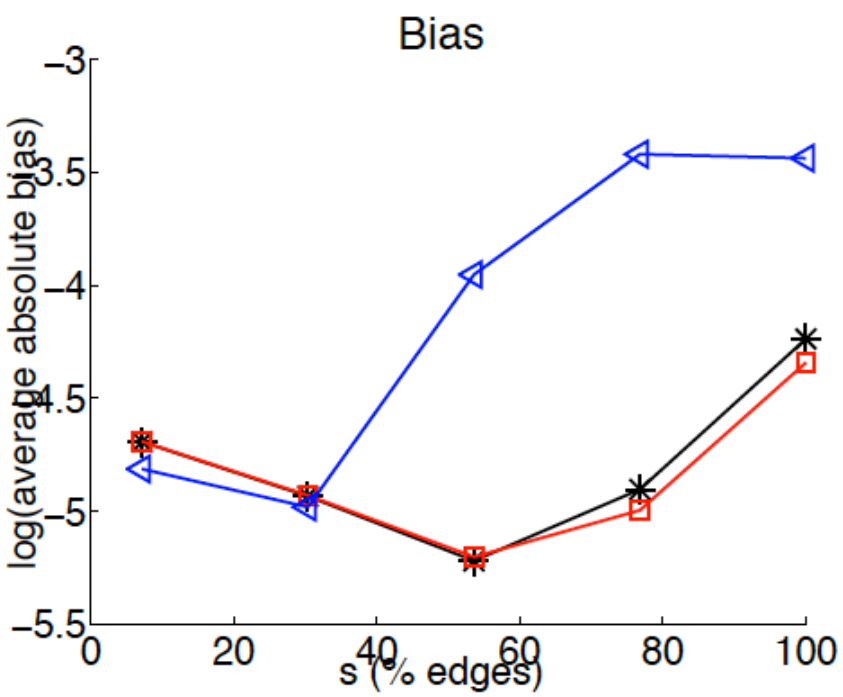
FO-grid-d Experiment (○,∗ , □, ◁ indicate ML-exact, PL, CD (K=5), PMM respectively)

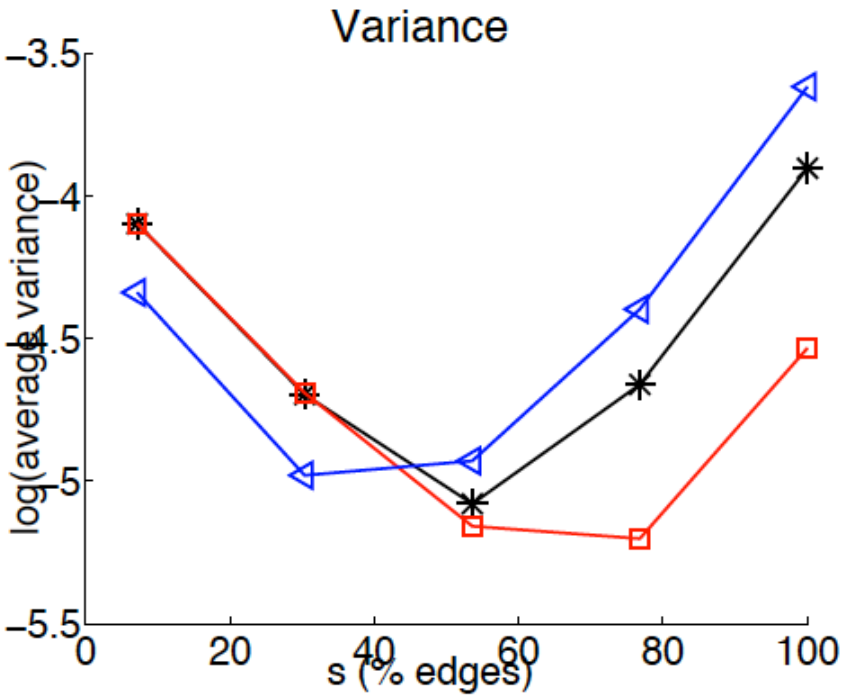
Some Experimental Results

Binary values (Boltzmann machine)
Fully observed
From 7x7 grid to complete graph by adding edges
Weights sampled from U[-0.1,0.1]
N = 10 x nr. parameters



Parise, Welling '05

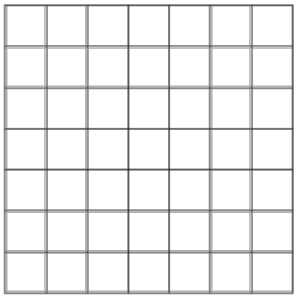




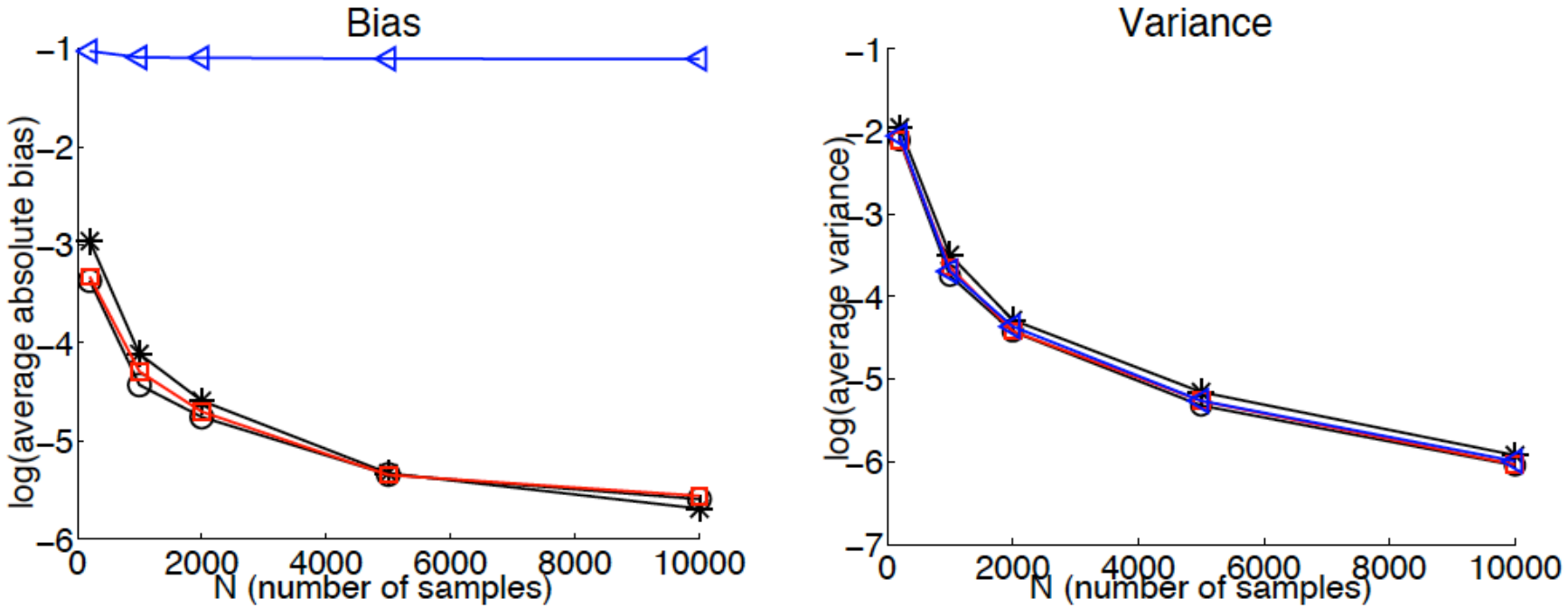
FO-FC-s Experiment (∗ , □, ◁ indicate PL, CD (K=5), PMM respectively)

Some Experimental Results

Parise, Welling '05



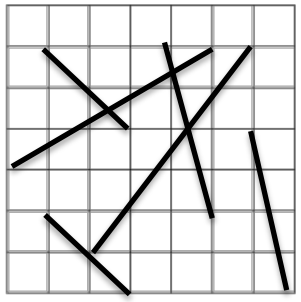
Binary values (Boltzmann machine)
Fully observed,
6x6 grid
N = varies
Real data, cropped 6x6 images from USPS digits
(first learn parameters using PL, then sample data from it)



FO-grid-N-real Experiment (○,∗ , □, ◁ indicate ML-exact, PL, CD (K=5), PMM respectively)

Some Experimental Results

Binary values (Boltzmann machine)
Fully observed
From 6x6 grid to complete graph by adding edges
N = 10 x nr. parameters
Real data from USPS digits
(first learn parameters using PL, then sample data from it)



Parise, Welling '05

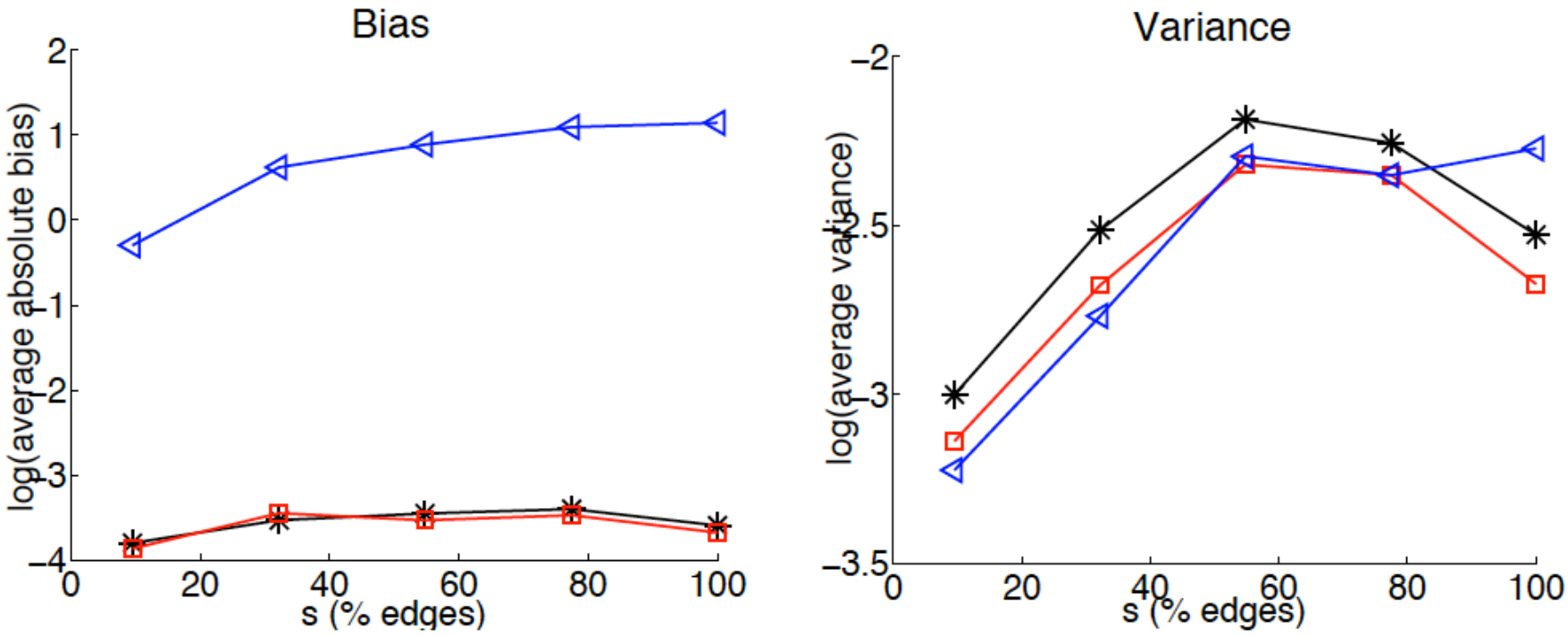
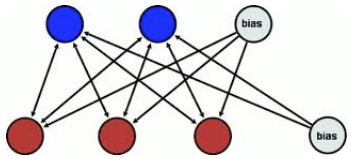


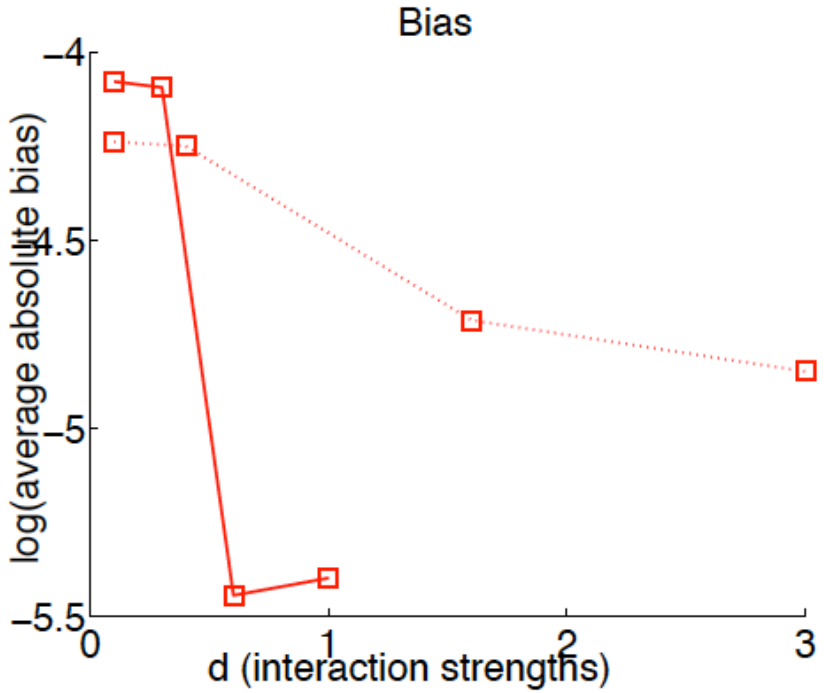
Figure 6: *FO-FC-s-real* Experiment (∗ , □, ◁ indicate PL, CD (K=5), PMM respectively)

Some Experimental Results

Binary values (Restricted Boltzmann machine)
10 Hidden and 10 Visible nodes all connected
OR 7x10 biparite grid (alternating visible hidden).
All interactions +ve sampled from U[0,d], vary d
N = 12000 (RBM), 20000 (Grid).
Initialize learning at true value (local minima)



Parise, Welling '05



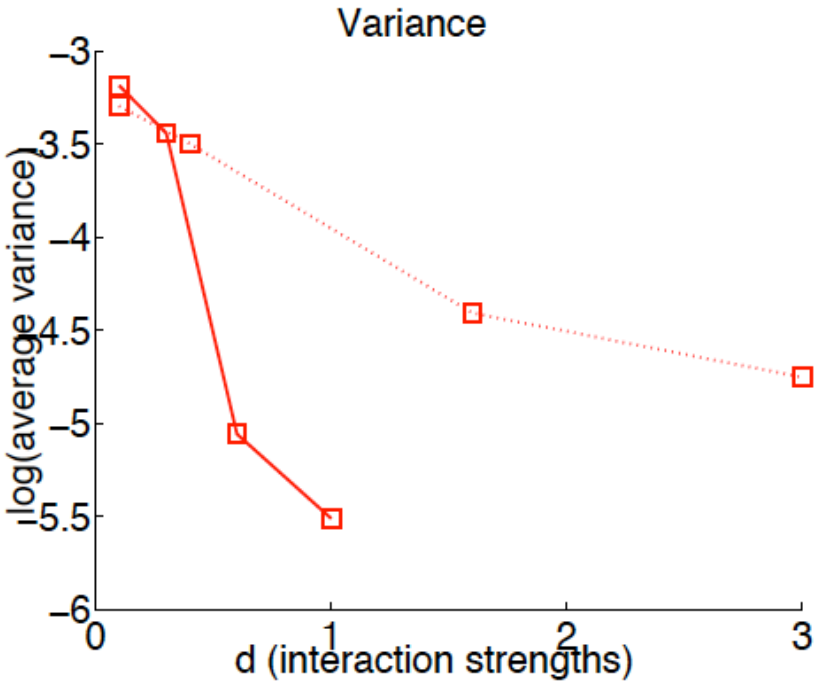
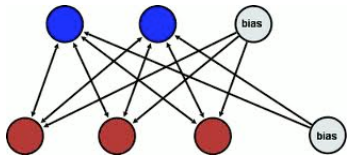


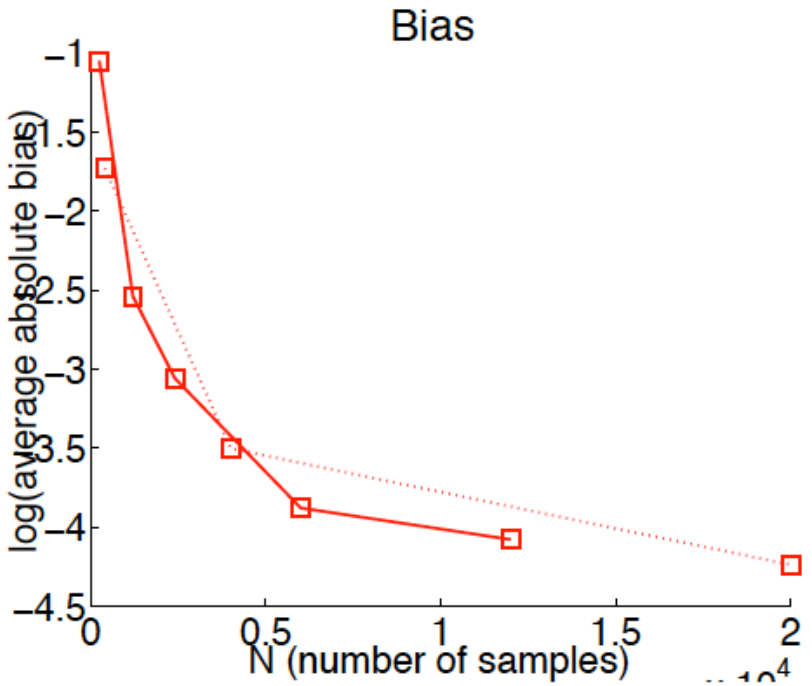
Figure 7: *PO-FC-d and PO-grid-d* Experiments (□ indicates CD (K=5). The solid line is for PO-FC-d and the dotted line is for PO-grid-d)

Some Experimental Results

Binary values (Restricted Boltzmann machine)
10 Hidden and 10 Visible nodes all connected
OR 7x10 biparite grid (alternating visible hidden).
All interactions +ve sampled from U[0,0.1]
Vary N
Initialize learning at true parameter values (local minima)



Parise, Welling '05



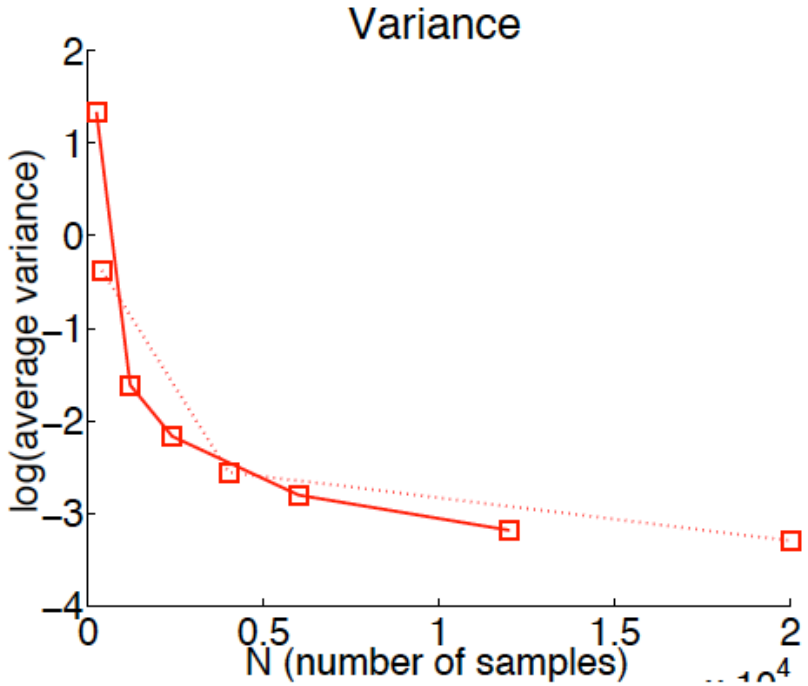
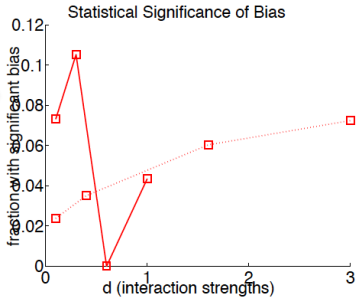


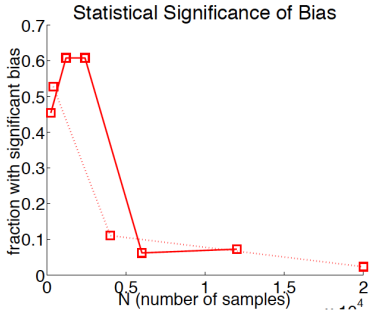
Figure 8: *PO-FC-N and PO-grid-N* Experiments (□ indicates CD (K=5). The solid line is for PO-FC-N and dotted line is for PO-grid-N)

Conclusions Experiments

- Fully Observed: PMM only works if interactions strength is small and sparsity is high.
- For FO problems, PL and CD worked equally well, but PL is faster.
- For Partially observed problems PMM and PL don't work, but CD works well.
- For PO problems (CD), the bias and variance decreased as interaction strength increased. Since variance decreases faster than bias, more parameters become *statistically significantly* biased.



- As N increases bias and variance decrease and the number of statistically significantly biased parameters also also decreases.



If model is fully observed: use PL
If model is partially observed: use CD

MCMC-MLE

Geyer '91

$$Z(\theta) = Z(\vartheta)\sum_X \exp\left[\sum_k (\theta_k - \vartheta_k) f_k(X_k)\right] q(X \mid \vartheta) \;\approx\; Z(\vartheta)\frac{1}{S}\sum_s \exp\left[\sum_k (\theta_k - \vartheta_k) f_k(x_{ks})\right]$$

- If we can easily sample from q(X), and q(X) is close to p(X) then this can be used to approximate Z.

- The gradient can be approximated as:

$$\nabla_{\theta_k} L = E_{\hat{P}}[f_k] - \frac{1}{S}\sum_s \left(\frac{\exp\left[\sum_k (\theta_k - \vartheta_k) f_k(x_{ks})\right]}{\frac{1}{S}\sum_{s'} \exp\left[\sum_k (\theta_k - \vartheta_k) f_k(x_{ks'})\right]} \right) f_k(x_{ks})$$

- As we update parameters, the weights quickly degenerate (effective sample size = 1).

Importance weights: $w_s = \frac{p(x_s \mid \theta)}{q(x_s \mid \vartheta)}$

- One can resample from the weights, and rejuvenate by running MCMC for some steps to improve the health of the particle set: *Particle Filtered MCMC* [Asuncion et al, 2010]

Composite Likelihood

Lindsey, '88

- We can easily generalize PL so that evaluate the probability at arbitrary blocks conditioned on (different) arbitrary blocks.
- This is called the composite likelihood: $CL = \sum_{i=1}^{n} \sum_{c=1}^{C} \log p(x_{iA_c} \mid x_{iB_c})$
- We want that at least every variable appears in some bock A_c.
  We want that for any particular component c, Ac and Bc are different.
  We want that A_c is not empty for every c.
- CL is still concave!
- Gradient: $\nabla_{\theta_k} CL = nE_{\hat{P}}[f_k] - \frac{1}{C_k} \sum_{c:X_{A_c} \cap X_k \neq 0} \sum_{i=1}^{n} E_{P(X_{A_c} \mid x_{i,B_c})}[f_k(X_{A_c} \cap X_k, x_{i,rest})]$
- We can now interpolate between ML and PL.

Persistent CD

Younes '89, Neal '92 Tieleman '08

- Instead of restarting the MCMC chains at the data-items, we can also initialize them at the last sample before the parameters were updated.

- This is exactly the same running Markov chains for some model P, and regularly interrupting these chains to change P a little bit.

- If the changes to P are small enough so that the chain can reach equilibrium between the updates, then the chains will track the distribution P and provide unbiased samples to compute the gradients.

- Example of a Robbins-Monro method to find a solution of: $E_{P_\theta}[f_k] - E_{\hat{P}}[f_k] = 0 \quad \forall k$ using a stochastic approximation method.

- This implies that we need a decreasing sequence of step-sizes such that:

$$\sum_{t=1}^{\infty} \eta_t = \infty, \quad \sum_{t=1}^{\infty} \eta_t^2 < \infty,$$

Herding

Welling '09

- Take the zero temperature limit: $\log Z(\frac{\theta}{T}) \xrightarrow{T \to 0} \frac{1}{T} \max_S \sum_k \theta_k f_k(S)$

- Then T*L0 converges to: $\ell \equiv TL_0 = \sum_k \theta_k E_{\hat{p}}[f_k] - \max_S \sum_k \theta_k f_k(S)$

- The gradient for some given state S is given as:

$$\nabla_{\theta_k} \ell = E_{\hat{p}}[f_k] - f_k(S)$$

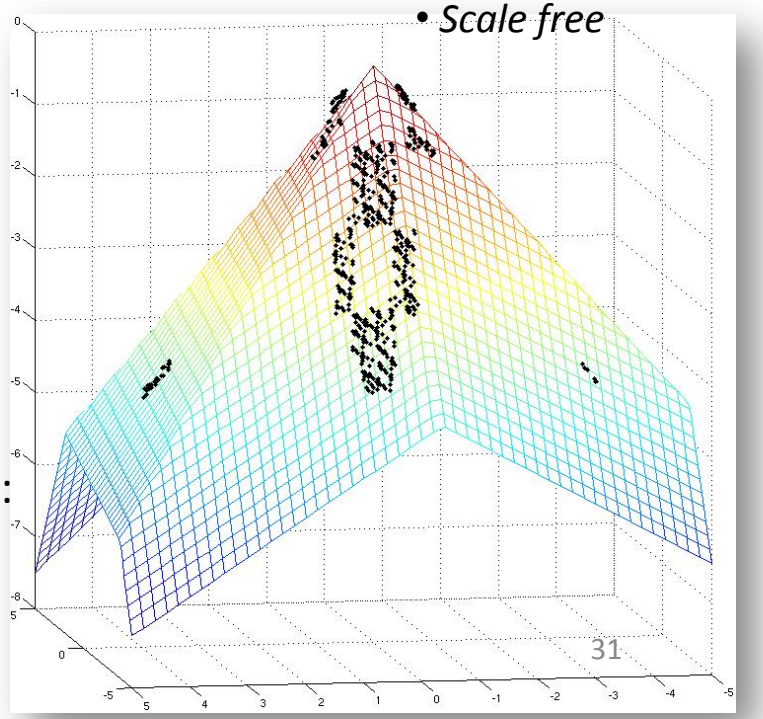
- Where: $S = \arg\max_S \left[\sum_k \theta_k f_k(S) \right]$

- Herding iterates these steps. It does not try to find the maximum (that's at 0), but it will generate pseudo-samples {St} that will satisfy the constraints:

$$E_{\hat{p}}[f_k] \approx \frac{1}{|S|} \sum_{t=1}^{|S|} f_k(S_t)$$

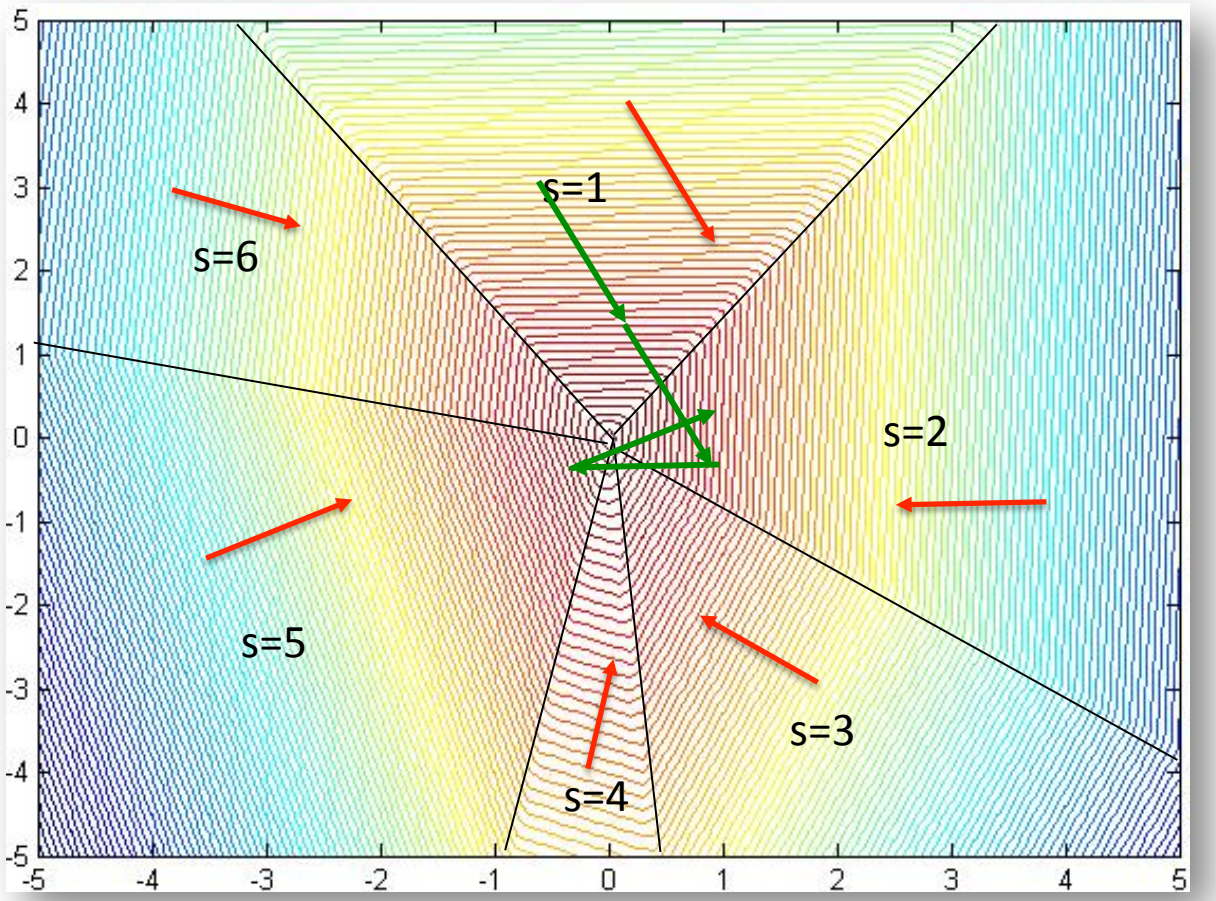
This function is:
- *Concave*
- *Piecewise linear*
- *Non-positive*
- *Scale free*

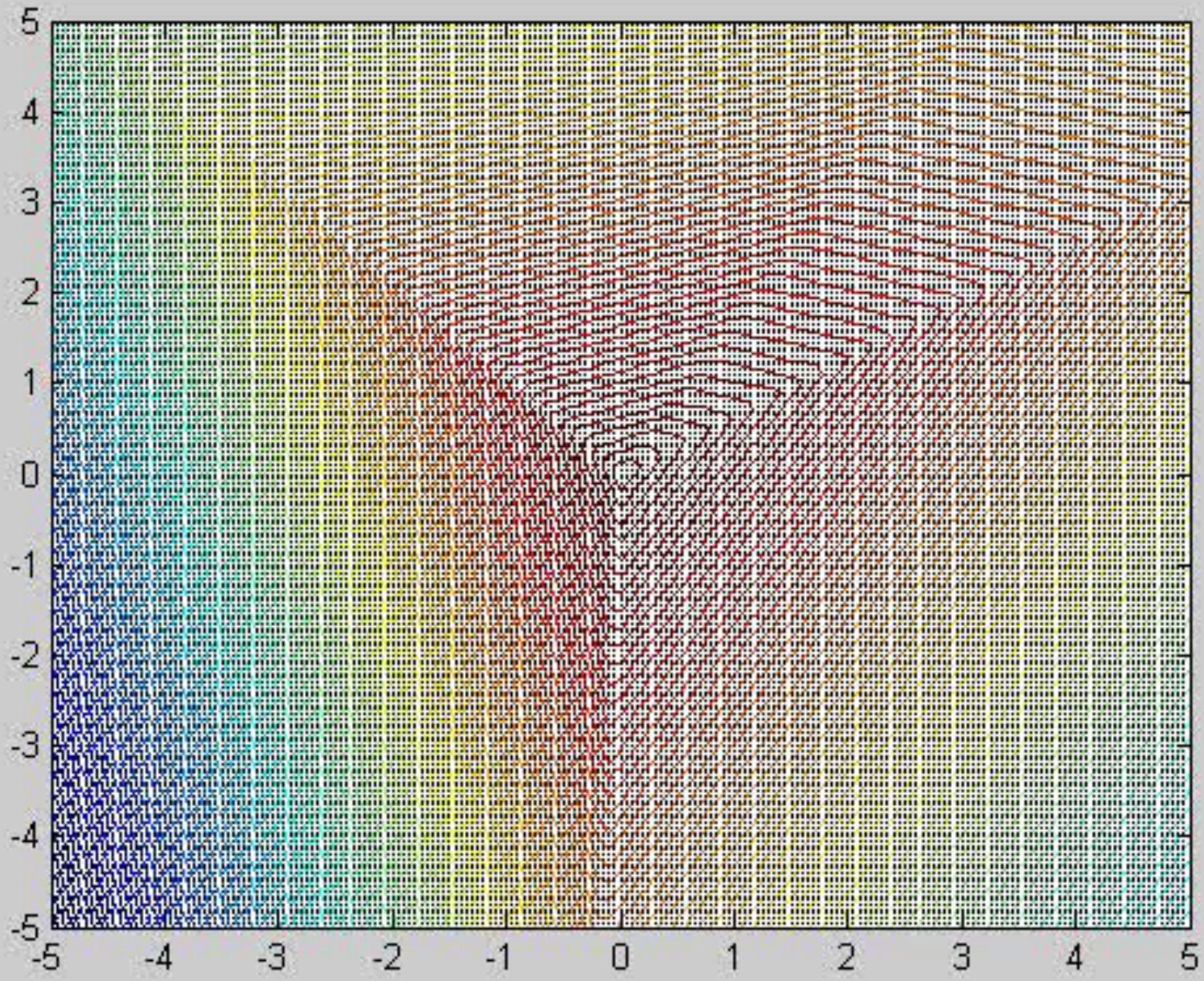
Tipi function:



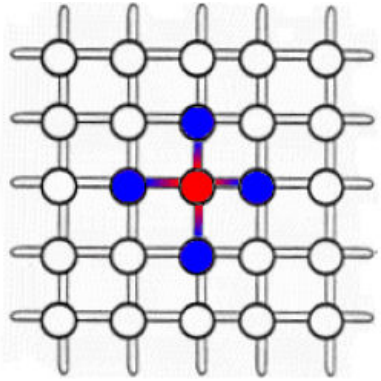
Example in 2-D

Itinerary: s=[1,1,2,5,2...




5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5

Grid



$$\sum_k w_k f_k(S) \Rightarrow \sum_{ij} w_{ij} s_i s_j + \sum_i w_i s_i$$

$$s_i^* = \delta\left[\sum_j w_{ij} s_j > -w_i\right]$$

Neuron fires if input exceeds threshold

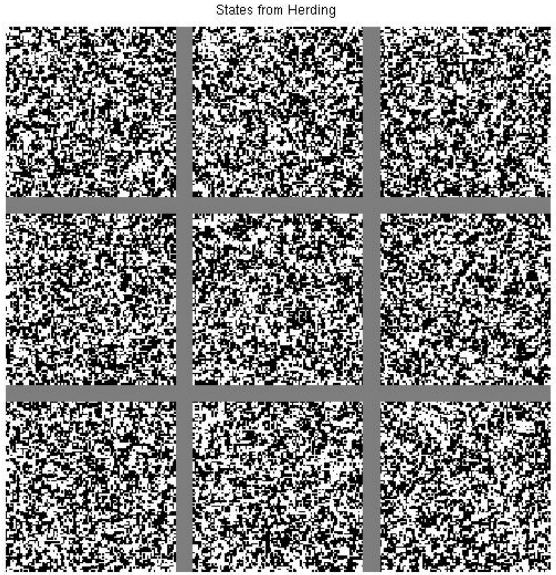
$$W_{ij} \leftarrow W_{ij} + E_{\hat{p}}[s_i s_j] - s_i^* s_j^*$$

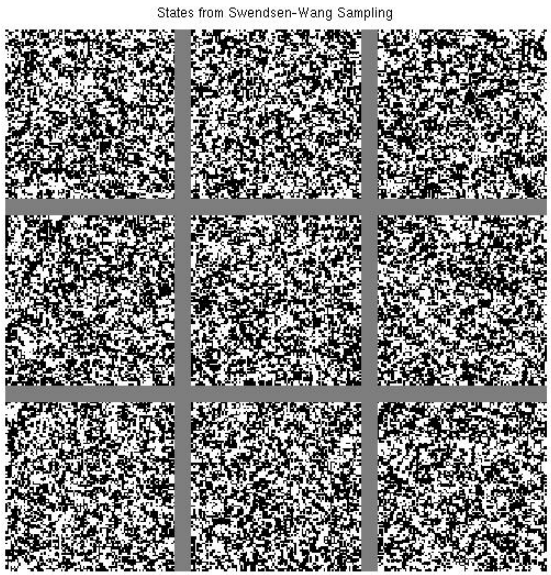
Synapse depresses if pre- & postsynaptic neurons fire.

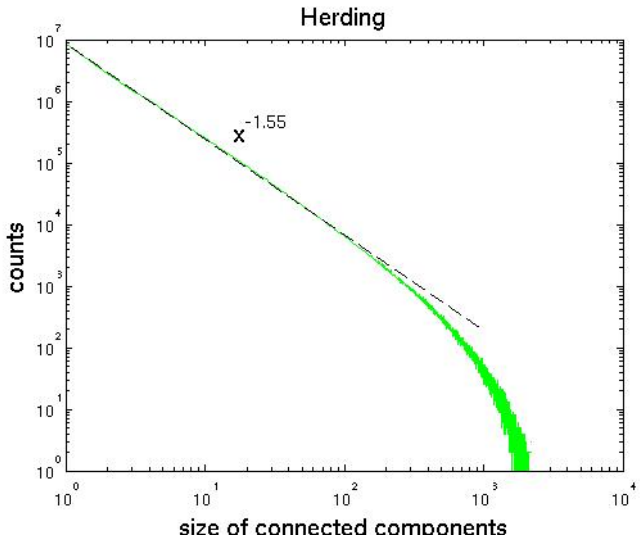
$$W_i \leftarrow W_i + E_{\hat{p}}[s_i] - s_i^*$$

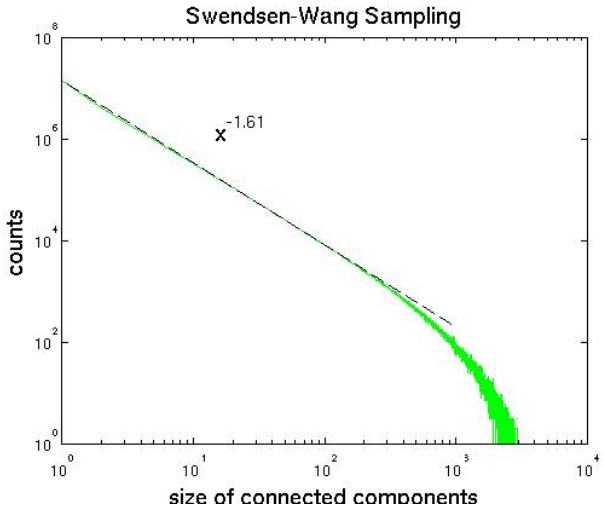
Threshold depresses after neuron fires

Pseudo-Samples From Critical Ising Model



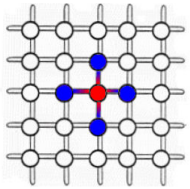






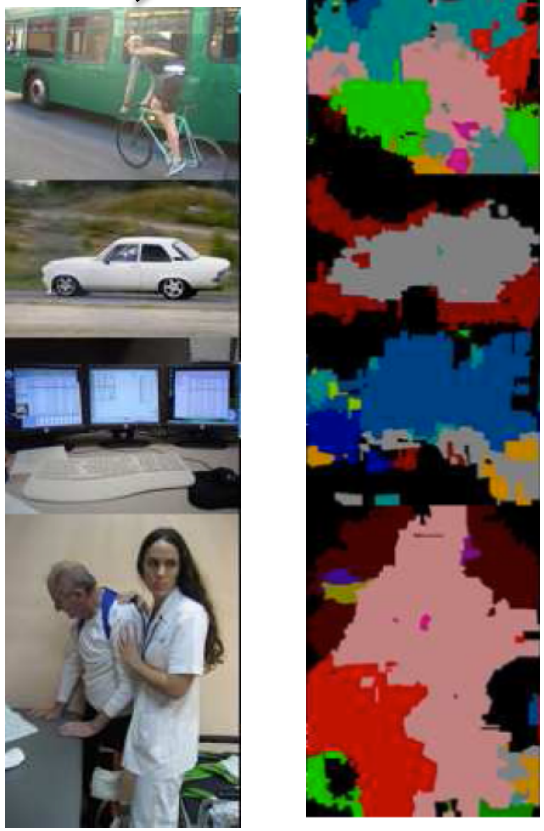
Example

Chen et al 2011



Classifier from local Image features:

$P(Object\ \ Category \mid Local\ \ Image\ \ Information)$

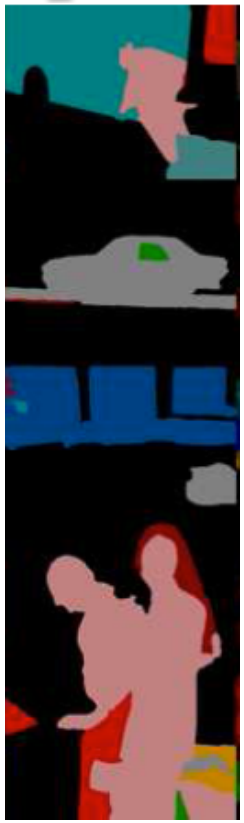


Combine with Herding

+

Classifier from boundary detection:

$P(Object\ \ Categories\ \ are\ \ Different\ \ across\ \ Boundary \mid Boundary\ \ Information)$



Herding will generate samples such that the local probabilities are respected as much as possible (project on marginal polytope)

Convergence

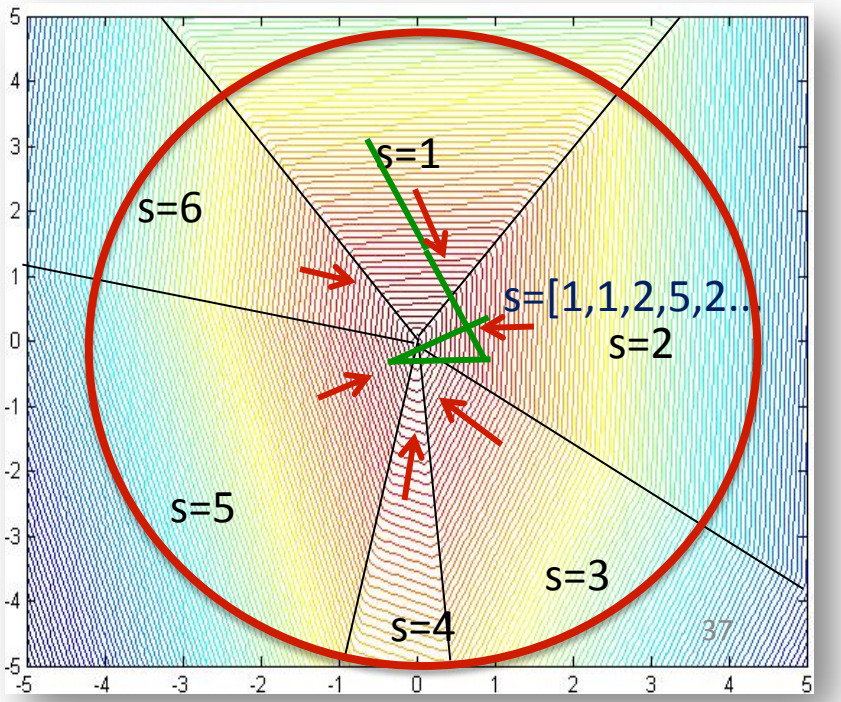
Translation: $v_t = E_{\hat{P}}[f] - f_k(S_t)$

Choose S_t such that: $\sum_k W_k v_k = \sum_k W_k \left(E_{\hat{P}}[f_k] - f_k(S) \right) \le 0$

Then: $|\frac{1}{T}\sum_{t=1}^{T} f_k(s_t) - E_{\hat{P}}[f_k]| \sim O(\frac{1}{T})$

Equivalent to "Perceptron Cycling Theorem" (Minsky '68)

Note: this implies that we do not have
To solve the MAP problem at every iteration
As long as the CPT holds!



Period Doubling

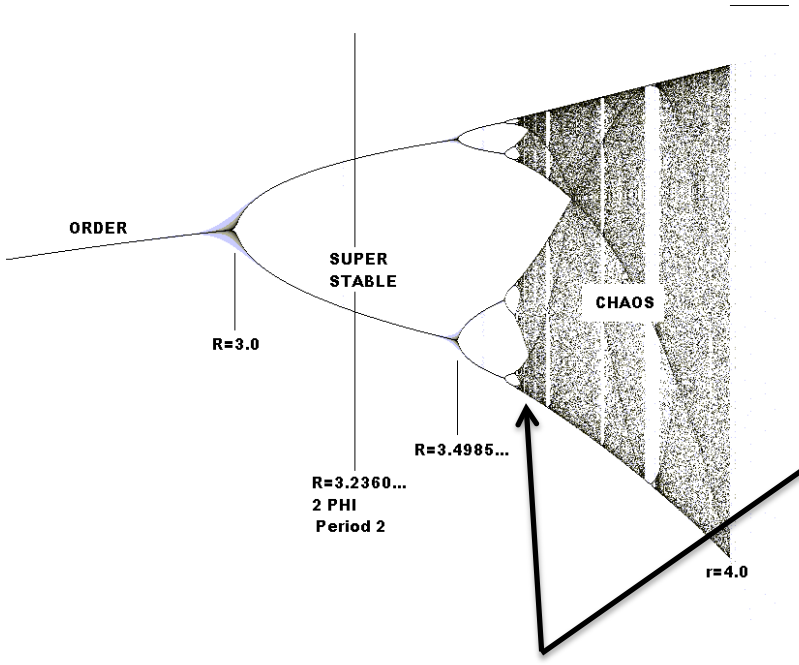
As we change R (T) the number of fixed points change.

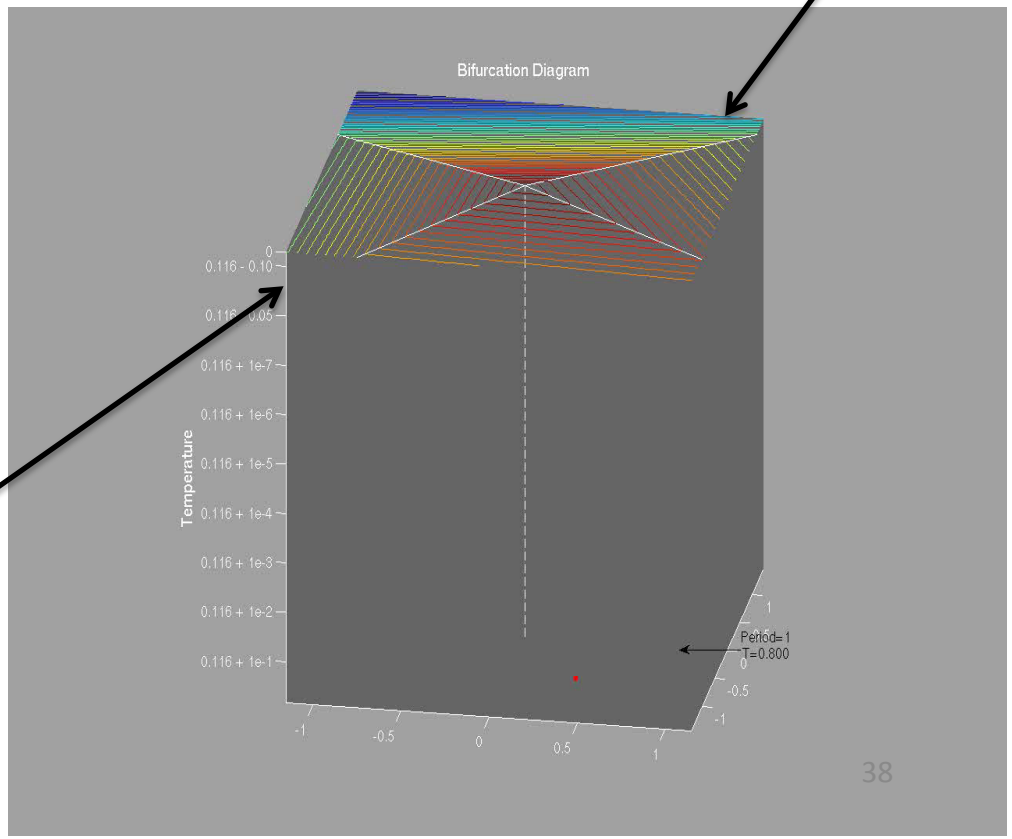
$$W_{k,t+1} \leftarrow W_{k,t} + \langle f_k \rangle - \frac{\sum_x f_k(x) \exp\left[\sum_{k'} \frac{W_{k',t}}{T} f_k(x)\right]}{\sum_x \exp\left[\sum_{k'} \frac{W_{k',t}}{T} f_k(x)\right]}$$

T=0: herding

Logistic map:

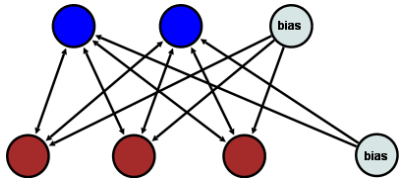
$$W_{t+1} = RW_t(1 - W_t)$$





"edge of chaos"

Herding with Hidden Units: RBM

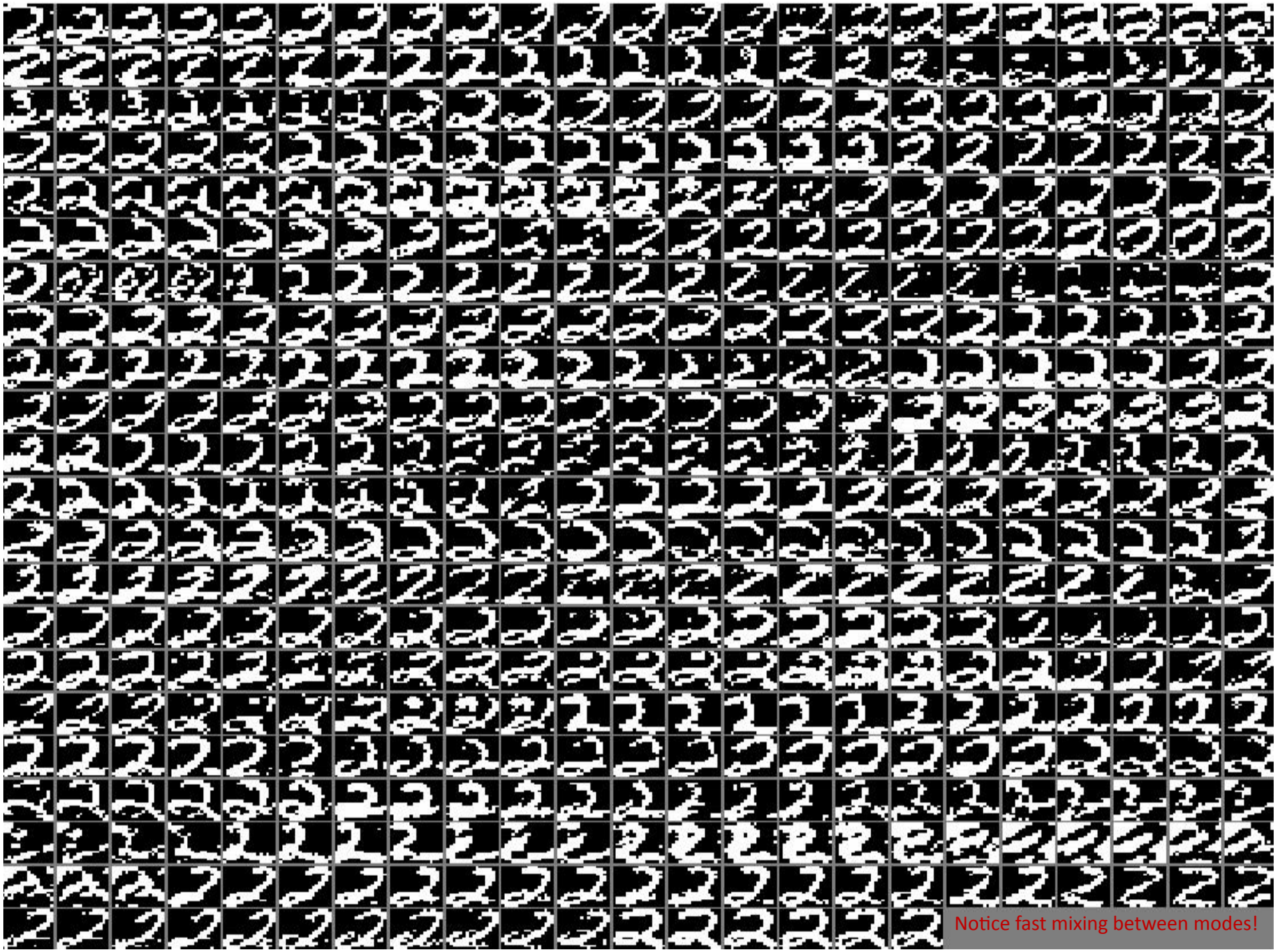


$$(Z^*, X^*) = \arg\max_{Z,X} \left[\sum_k W_k f_k(Z, X) \right]$$

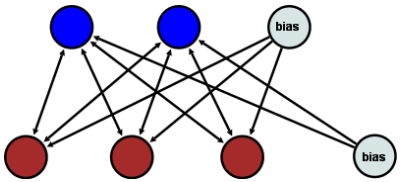
$$\hat{Z}_n = \arg\max_{Z_n} \left[\sum_k W_k f_k(Z_n, X_n) \right]$$

$$W_k \leftarrow W_k + \frac{1}{N} \sum_n f_k(\hat{Z}_n, X_n) - f_k(Z^*, X^*)$$

inputation through maximization


Notice fast mixing between modes!

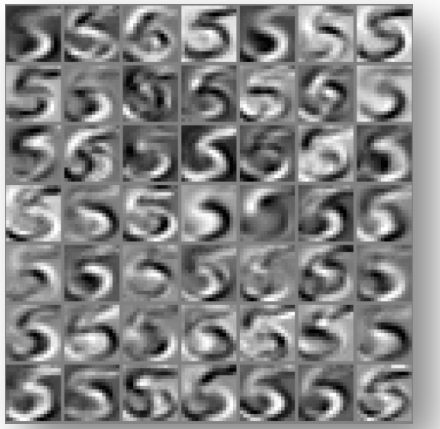
A New Learning Rule

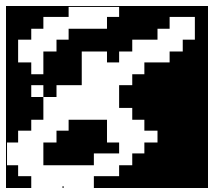


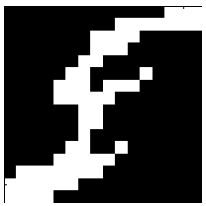
$$\left\langle x_i z_j \right\rangle_{t+1} = (1-\gamma_t)\left\langle x_i z_j \right\rangle_t + \frac{\gamma_t}{|S_t|}\sum_{n\subset S_t} x_{in}\hat{z}_{jnt}$$

$$W_{ij,t+1} = W_{ij,t} + \left\langle x_i z_j \right\rangle_t - x^*_{it} z^*_{jt}$$

RBM → Hopfield net
Runs without the data.







Dreaming about digits

Max-Margin Markov Networks

Taskar et al '02

- If we don't care about the probabilities but we do care about making the right decision, then there is a "max-margin" alternative to CRFs.

- On the training data we require that the energy of the true label (given the attributes) is lower than the energy of any other possible state , by pre-specified margin.

$$\min_{\{\theta_k\}} \| \theta \|^2 + C \sum_i \xi_i$$

$$s.t. \quad \sum_k \theta_k \left[f_k(y_{ki}, x_{ki}) - f_k(Y_k, x_{ki}) \right] \geq \Delta(Y, y_i) - \xi_i \qquad \forall i, \ \forall Y \neq y_i$$

- The margin depends on (Y, y_i) so that almost correct states Y have a smaller margin.

- Optimization looks daunting because of exponentially many constraints. However, we can add the constraints one-by-one (constraint generation). Often, you are done after you added a polynomial nr. of constraints.

- To find which constraints to add, you run MAP-inference on the current QP problem for every data-case. If MAP states satisfy constraints you are done.

Structure Learning: L1-regularization

Lee et al '06, Abbeel et al '06

- How can we decide which features to include in the model?
- One practical way is to add L1-regularization to the parameters:

$$Score(M_K) = L_{M_K}(\theta^{MAP}) - \lambda \sum_{k=1}^{K} | \theta_k^{MAP} |$$

- The L1 penalty will drive unimportant features to 0, effectively removing them from the model.
- We can thus iteratively add features to the model until the score no longer increases.
- There is a PAC-style bound to guarantee that the solution is close to the best solution (in a KL-sense).

Bayesian Methods

- "Doubly intractable" because of partition function and Bayesian integration over parameters.
- Even MCMC would require one to compute Z for the old and proposed state at every iteration.
- A nifty algorithm exists that circumvents the computation of Z if we can draw perfect samples from P(X). [Moller et al '04, Murray et al. '06]
- We will discuss approximate MCMC & Laplace approximations. There is also an approach for CRFs based on EP [Qi et al '05].
- Without hidden units the posterior is unimodal.
- With hidden units the problem is "triply intractable" because a very large number of modes may appear in the posterior.

Bayesian Methods (1)
Approximate Langevin Sampling

[Murray, Ghahramani '04]

- MCMC with accept/reject requires the computation of Z for every sample drawn.
- However, Langevin sampling does not requires this: $\theta_{t+1} = \theta_t + \frac{\eta^2}{2}\nabla_\theta L(\theta_t, x) + \eta\varepsilon_t \qquad \varepsilon_t \sim N[0, I]$
  (catch: this only samples correctly when $\eta \to 0$) .
- Idea: Let's try for finite stepsizes. However, we still need the gradient.
  Idea: Let's use contrastive divergence gradients (brief MCMC chains started at the data).
- This approximate sampling scheme worked well empirically.
  (gradients based on mean field did not work, gradients based on BP sometimes worked).

Bayesian Methods: Laplace + BP

Parise & Welling '06, Welling & Parise '06

$$p(x) = \int d\theta \;\; p(x \mid \theta) p(\theta)$$

$$= \sum_{i=1}^{n} \sum_{k} \theta_k^{MAP} f_k(x_{ki}) - n \log Z(\theta^{MAP}) - \frac{1}{2} K \log(n) + \log p(\theta^{MAP}) + \frac{1}{2} K \log(2\pi) - \frac{1}{2} \log \det \left[C - \frac{\Lambda}{n} \right]$$

- Unlike MCMC, we only have to compute Z once at the MAP state.
  We can approximate Z using "Annealed Importance Sampling" or the Bethe free energy.

- C is the covariance between all fairs of features:

$$C_{kl} = E_{p(X_k \cup X_l \mid \theta^{MAP})}[f_k f_l] - E_{p(X_k \mid \theta^{MAP})}[f_k] E_{p(X_l \mid \theta^{MAP})}[f_l]$$

- Approximate log(Z) with Bethe free energy, and C using "linear response propagation"
  LRP is an additional propagation algorithm after BP has converged.
  For Bolzmann machine there is an analytic expression available.

[Welling, Teh '03]

- Similar expression for CRFs.

Experiments: Methods (1)

$$MAP = \sum_{i=1}^{n} \sum_{k} \theta_k^{MAP} f_k(x_{ki}) - n \log Z(\theta^{MAP}) + \log p(\theta^{MAP})$$

$$BIC - ML = \sum_{i=1}^{n} \sum_{k} \theta_k^{MAP} f_k(x_{ki}) - n \log Z(\theta^{MAP}) - \frac{1}{2} K \log(n)$$

$$BP - LR = \sum_{i=1}^{n} \sum_{k} \theta_k^{MAP} f_k(x_{ki}) - n \log Z(\theta^{MAP}) - \frac{1}{2} K \log(n) + \log p(\theta^{MAP}) + \frac{1}{2} K \log(2\pi) - \frac{1}{2} \log \det \left[C - \frac{\Lambda}{n} \right]$$

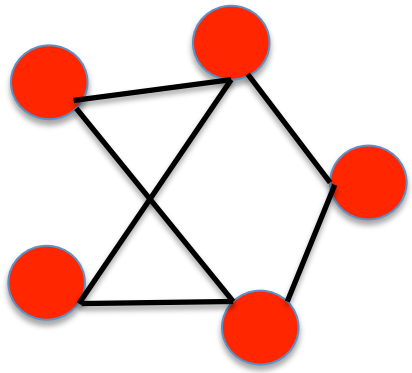
BP-LR-ExactGrads: BP-LR but using exact gradient to find the MAP-value of the parameters.

Laplace-Exact: Laplace approximation, but computing the MAP value exactly and computing the covariance "C" and the partition function "Z" exactly.

Annealed Importance Sampling = "ground truth": [Neal 2001].
MCMC approach to compute evidence.
This requires the exact evaluation of the Z at every iteration.

Experiments: Methods (2)

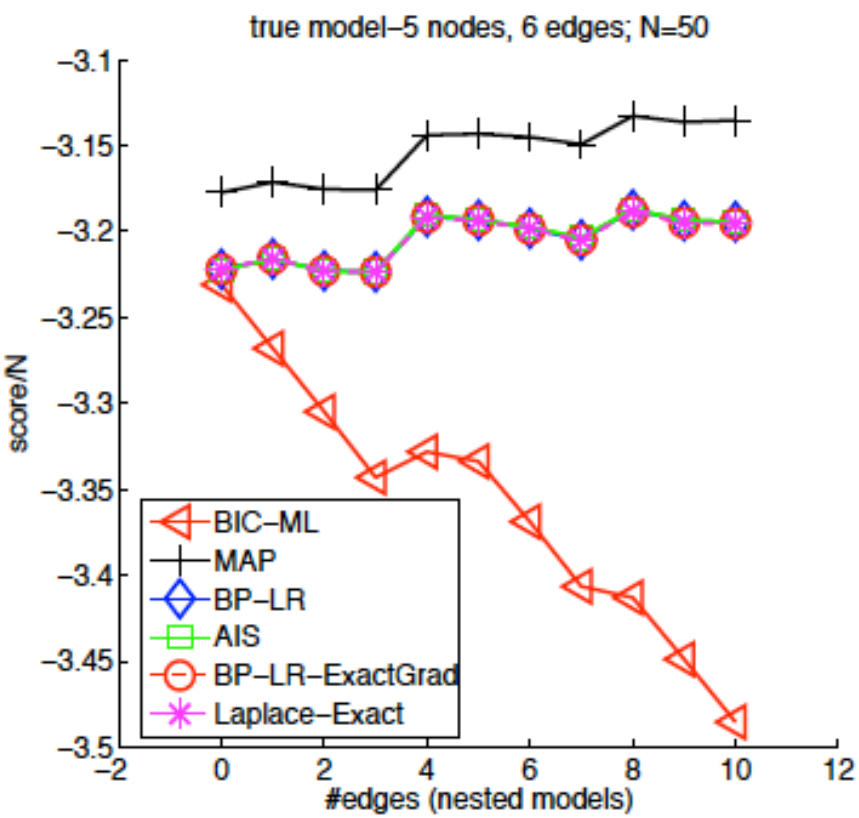
We generated 50 different random structures on 5 nodes. For each we sample 6 different sets of parameters with weights $w \sim \mathcal{U}\{[-d, -d+\epsilon] \cup [d, d+\epsilon]\}, d > 0, \epsilon = \frac{0.1}{4}$ and biases $b \sim \mathcal{U}[-1, 1]$ and varying the edge strength d in $[\frac{0.1}{4}, \frac{0.2}{4}, \frac{0.5}{4}, \frac{1.0}{4}, \frac{1.5}{4}, \frac{2.0}{4}]$. We then generated $N = 10000$ samples from each of these (50×6) models using exact sampling by exhaustive enumeration.

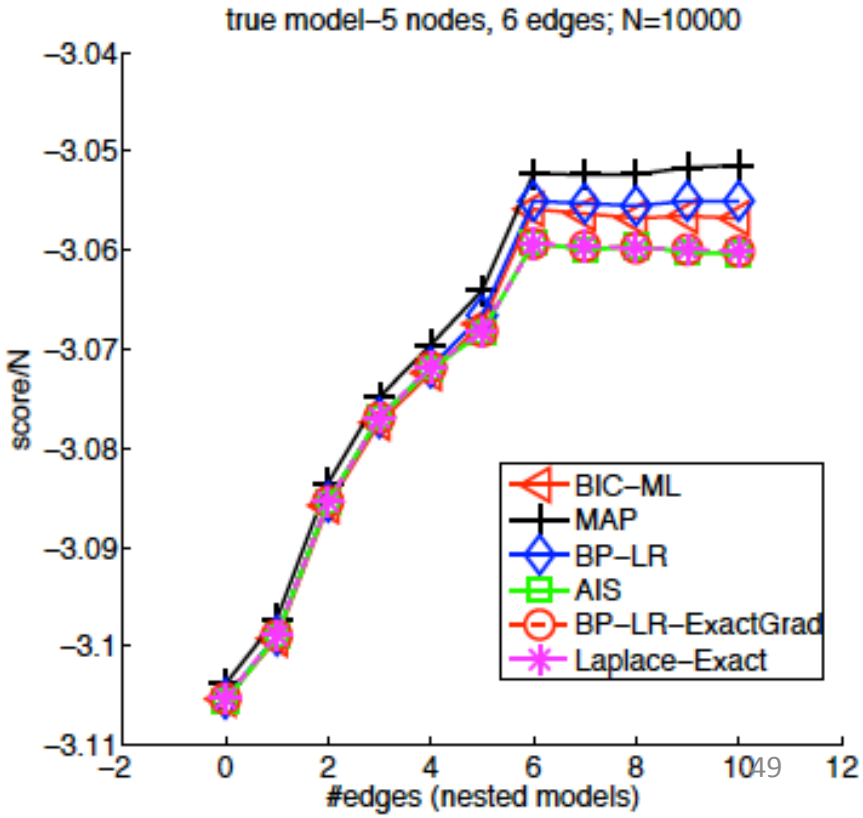


We will vary:
d = edge strength
N = sample size
Nr. edges

Experiments: Results (1)

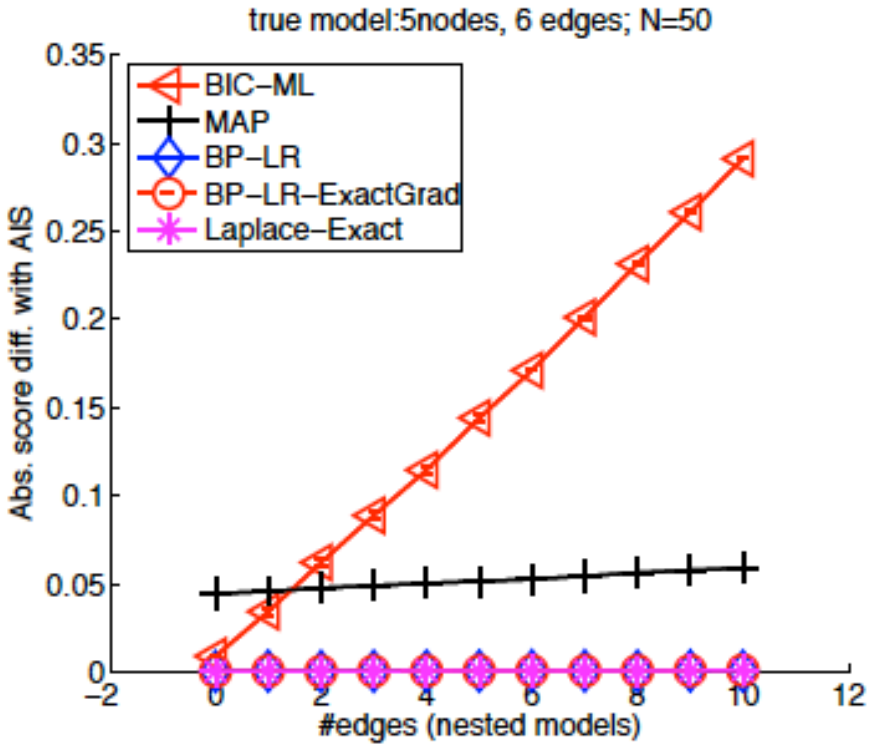
Vary the number of edges in 5 node model. (d=0.5/4)

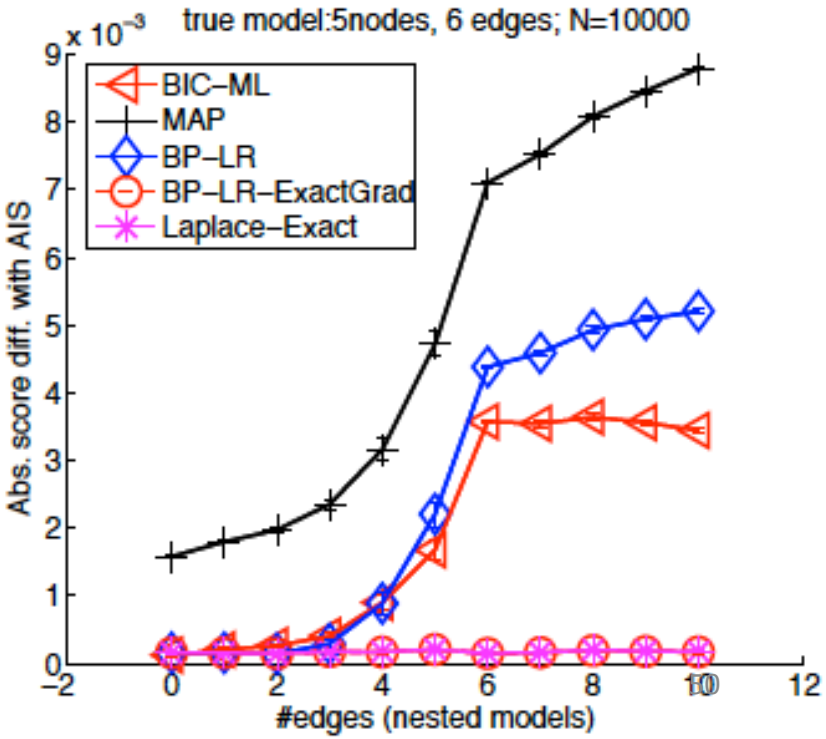




Experiments: Results (2)

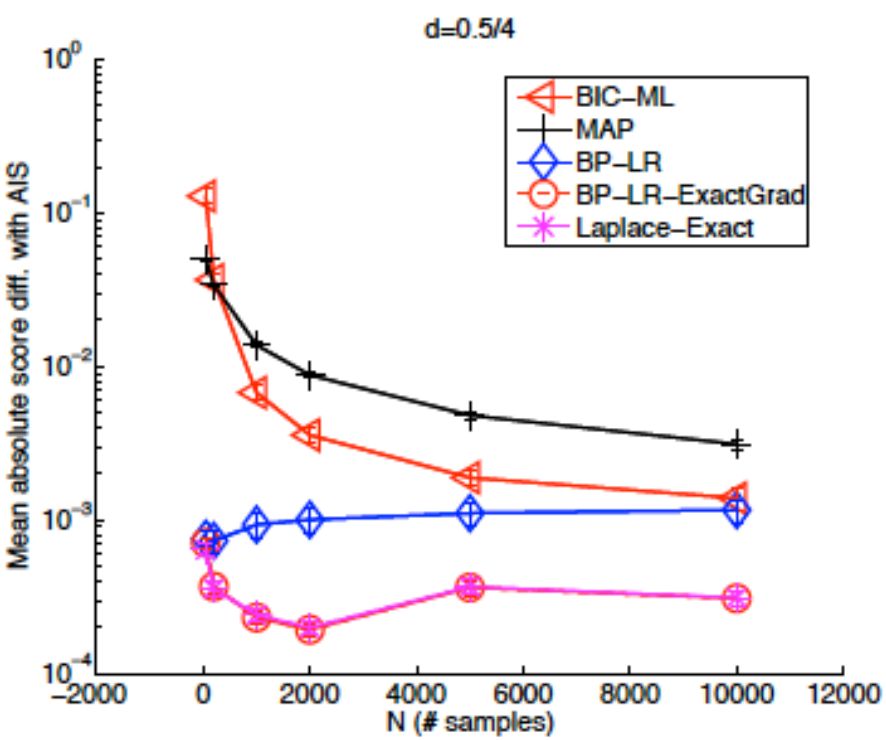
Same experiment, but we plot difference with AIS (d=0.5/4)

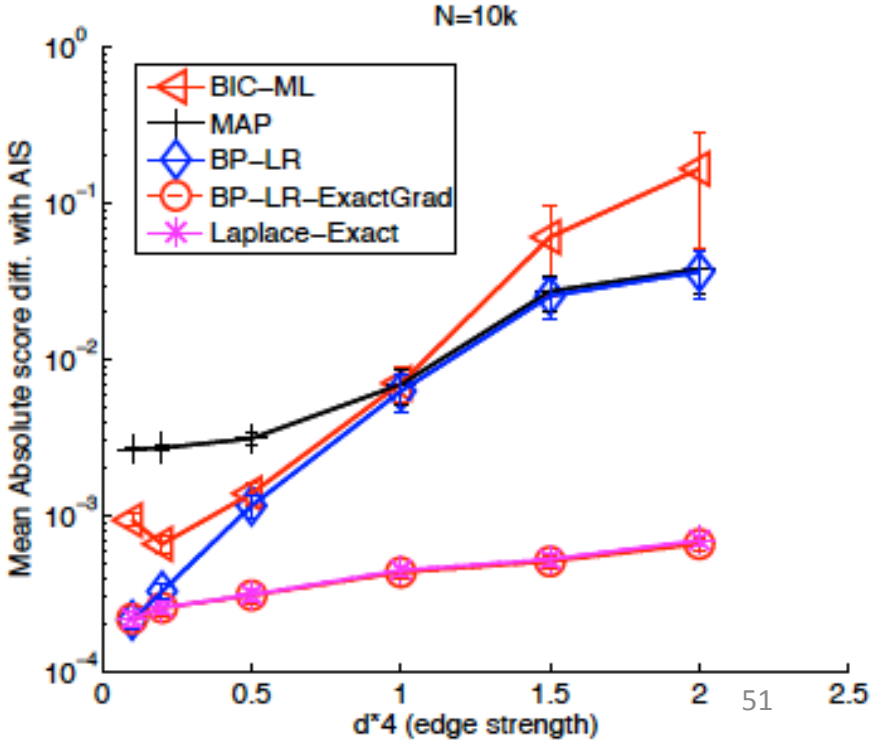




Experiments: Results (3)

Vary nr samples N (left) and edge strength d (right).





Experiments: Conclusion

- Laplace approximation can be very accurate relative to Map and BIC methods for small N. This means that the logdet(C) term is very important (even though it is O(1)).

- BP-LR deteriorates when N gets larger (unlike MP and BIC which get more accurate). Reason: gradient descend with BP can not locate MAP value accurately relative to the width of the posterior → Laplace approximation gets bad. Solution: use CD to find MAP value and AIS to compute Z (requires running AIS once).

- When interaction strengths increase all methods deteriorate.

Conclusions

	MRF	CRF	PO-MRF
Learning	1. Singly Intr.	2.	3.
Bayesian Learning	4. Doubly Intr.	5.	6. Triply Intr.

1. partition function intractable, likelihood concave
2. partition function for every data-case, likelihood concave
3. partition function for every data-case, likelihood many local maxima
4. integration over parameters + intractable Z, unimodal posterior
5. separate integrations for every data-case + intractable Z, unimodal posterior
6. integration over hidden units & parameters + intractable Z, multimodal posterior

Quiz: What do the following stand for?

- Learning: PMM, PL, CL, CD, PCD, MCMC-MLE, MMMN, Herding
- Structure: L1-LL, Langevin+CD, Laplace+BP