Relaxations for inference in restricted Boltzmann machines

Sida Wang* Roy Frostig* Percy Liang Christoper D. Manning
{sidaw, rf}@cs.stanford.edu

ICLR 2014

April 15, 2014

1. Background
   - The problem
   - Integer quadratic program
2. Randomized Relax and Round
   - Relaxations
   - Randomized rounding
3. Evaluations
4. Conclusions

Outline

1 Background
- The problem
- Integer quadratic program

2 Randomized Relax and Round
- Relaxations
- Randomized rounding

3 Evaluations

4 Conclusions

ICLR context

Probablistic models seems to be losing to feedforward networks.

Probablistic models / max-margin models requiring inference are still necessary/better at tasks having structured outputs:

- Word alignment
- CRFs for image segmentation, sequence tagging
- Parsing

or if you want to make use of unlabelled data.

Inference methods

Partly because we cannot do inference as well as we can do gradient descend

- MCMC (Gibbs)
- Variational inference (mean field)
- Belief propagation (not used in RBMs)
- Relaxation (not used in RBMs)

We propose a new relaxation-based inference method and solve it using gradient descend. Applicable to RBMs, DBMs, MRFs, CRFs.

Theoretical properties

Inference Method	Terminate	Correct
MCMC	No	Yes
practical MCMC	Yes	No
variational inference	Yes	No
belief prop.	No	No
relaxation	Yes	Approx.

Terminate: guaranteed convergence to an answer (say in polytime)
Correct: guaranteed to return the right answer

Theoretical properties

Inference Method	Terminate	Correct
MCMC	No	Yes
practical MCMC	Yes	No
variational inference	Yes	No
belief prop.	No	No
relaxation	Yes	Approx.

Terminate: guaranteed convergence to an answer (say in polytime)
Correct: guaranteed to return the right answer
Approx.: within some multiplicative/additive constant of the right answer

Theoretical properties

Inference Method	Terminate	Correct
MCMC	No	Yes
practical MCMC	Yes	No
variational inference	Yes	No
belief prop.	No	No
relaxation	Yes	Approx.

Terminate: guaranteed convergence to an answer (say in polytime)
Correct: guaranteed to return the right answer
Approx.: within some multiplicative/additive constant of the right answer
Cannot have 2 yes because Long and Servedio, 2010 showed it's NP hard to do inference even in RBMs, even approximately.

How to make practical use of a hardness result

To show problem A is hard:

- Find a similar problem B known to be hard
- Make enough assumptions on A so that B reduces to A

How to make practical use of a hardness result

To show problem A is hard:

- Find a similar problem B known to be hard
- Make enough assumptions on A so that B reduces to A

To make use of the hardness proof:

- Look at the assumptions needed to make A harder than B
- Decide that B is harder than practical instances of A
- Try to solve A using the best approximation algorithm for B

How to make practical use of a hardness result

To show problem A is hard:

- Find a similar problem B known to be hard
- Make enough assumptions on A so that B reduces to A

To make use of the hardness proof:

- Look at the assumptions needed to make A harder than B
- Decide that B is harder than practical instances of A
- Try to solve A using the best approximation algorithm for B

B is CUTNORM (Alon and Naor, 2006), which is a special case of MAXCUT.

What is this work?

Do RBM inference by the (almost provably) optimal approximation algorithm for MAXCUT

- SDP relaxation and randomized rounding from vectors
- solve a practical low-rank version of the SDP
- use random roundings to get a variety of near-MAP solutions

Flash RBM review

The restricted Boltzmann Machine defines a probability distribution on bit vector $v \in \{0,1\}^n$

- RBM:
$$p_W(v) = \frac{1}{Z} \sum_h \exp(v^\top W h + a^\top v + b^\top h)$$
- Energy $E(v,h) = -v^\top W h - a^\top v - b^\top h$
- Partition function $Z = \sum_{v,h} \exp(v^\top W h + a^\top v + b^\top h)$
- Bipartite Markov random field with latent variables
- Better guarantees here, but we present a more general solution.

MAP as an IQP

MAP inference in RBM is an instance of the integer quadratic program

$$\begin{array}{ll} \text{maximize} & x^\top A x \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

- $x^\top A x = \sum_{i,j} A_{ij} x_i x_j$

MAP as an IQP

MAP inference in RBM is an instance of the integer quadratic program

$$\begin{array}{ll} \text{maximize} & x^\top A x \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

- $x^\top A x = \sum_{i,j} A_{ij} x_i x_j$
- For the RBM

$$A = \frac{1}{2} \begin{bmatrix} 0 & W \\ W^\top & 0 \end{bmatrix} \qquad\qquad x = [v, h]^\top$$

MAP as an IQP

MAP inference in RBM is an instance of the integer quadratic program

$$\begin{array}{ll} \text{maximize} & x^\top A x \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

- $x^\top A x = \sum_{i,j} A_{ij} x_i x_j$
- For the RBM

$$A = \frac{1}{2} \begin{bmatrix} 0 & W \\ W^\top & 0 \end{bmatrix} \qquad\qquad x = [v, h]^\top$$

- The case $x \in \{0, 1\}^n$ and any biases can also be reduced to a different A.

Outline

1. Background
   - The problem
   - Integer quadratic program
2. Randomized Relax and Round
   - Relaxations
   - Randomized rounding
3. Evaluations
4. Conclusions

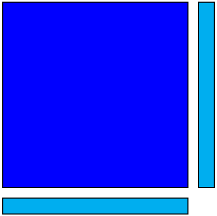
Warm-up: a simple relaxation

MAP inference in RBM is an instance of the integer quadratic program

$$\begin{array}{ll} \text{maximize} & x^\top A x \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

Relaxes to

$$\begin{array}{ll} \text{maximize} & x^\top A x \\ \text{subject to} & x \in [-1, 1]^n \end{array}$$



Introducing the SDP relaxation

MAP inference in RBM is an instance of the integer quadratic program

$$\begin{array}{ll} \text{maximize} & x^\top A x = \mathrm{tr}(A x x^\top) \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

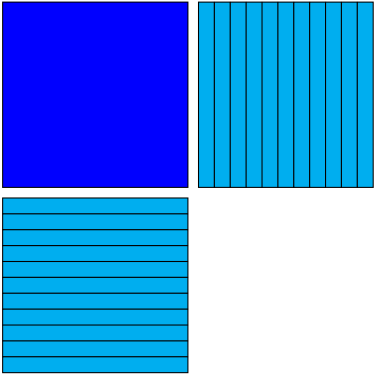
Make the objective linear by reparametrizing $S = xx^\top$

$$\begin{array}{ll} \text{maximize} & \mathrm{tr}(A x x^\top) = \mathrm{tr}(AS) \\ \text{subject to} & S \in \mathbb{R}^{n \times n} \\ & S \succeq 0,\ \mathrm{diag}(S) = 1 \\ & \mathrm{rank}(S) = 1 \end{array}$$

Dropping the non-convex constraint $\mathrm{rank}(S) = 1$ gives us a *semidefinite program* (SDP).

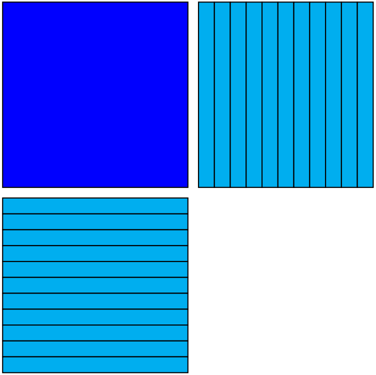
SDP as relaxation of IQP

Picture: SDP



SDP as relaxation of IQP

Picture: SDP



Rows have Euclidean norm 1

Rank k relaxation

While the SDP can be solved efficiently in theory, it does not scale very well in n. Consider the rank k relexation instead:

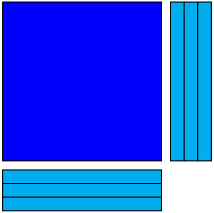
$$\begin{array}{ll} \text{maximize} & \operatorname{tr}(AS) \\ \text{subject to} & S \in \mathbb{R}^{n\times n} \\ & S \succeq 0,\ \operatorname{diag}(S) = 1 \\ & \operatorname{rank}(S) \leq k \end{array}$$

We can reparametrize in the reverse $S = XX^\top$:

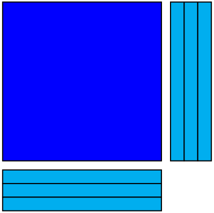
$$\begin{array}{ll} \text{maximize} & \operatorname{tr}(AS) = \operatorname{tr}(X^\top AX) \\ \text{subject to} & X \in \mathbb{R}^{n\times k} \\ & \text{for the i-th row: } ||X_i||_2 = 1 \end{array}$$

We lose convexity, but we can efficiently find a local minimum by projected gradient descend on X.

A picture



A picture



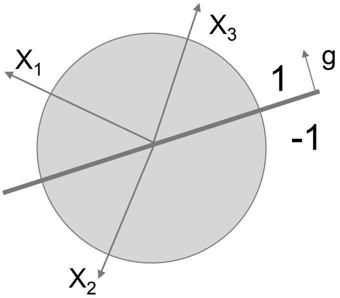
Rank 3 relaxation. Rows have Euclidean norm 1.

$X \to x$ rounding scheme

Goemans and Williamson style rounding for MAX-CUT

Given $X \in \mathbb{R}^{n \times k}$, we want to get $x \in \{-1, 1\}^n$:

- Sample random spherical unit vector g
- Take each $x_i = \mathrm{sgn}(\langle X_i, g \rangle)$



Outline

1. Background
   - The problem
   - Integer quadratic program
2. Randomized Relax and Round
   - Relaxations
   - Randomized rounding
3. Evaluations
4. Conclusions

MAP Inference in RBM

Solving

$$\begin{array}{ll} \text{maximize} & x^\top A x = \mathrm{tr}(A x x^\top) \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

with randomized relax and rounding (rrr)

	rrr	AG	rrr-AG	Gu
MNIST	340.29	377.47	377.39	319.34
Random	22309	22175	**23358**	12939
Hard	40037	36236	**41016**	23347

- AG: annealed Gibbs
- Gu: Gurobi IQP solver, given 10x longer time

RRR can be faster and better than annealed Gibbs

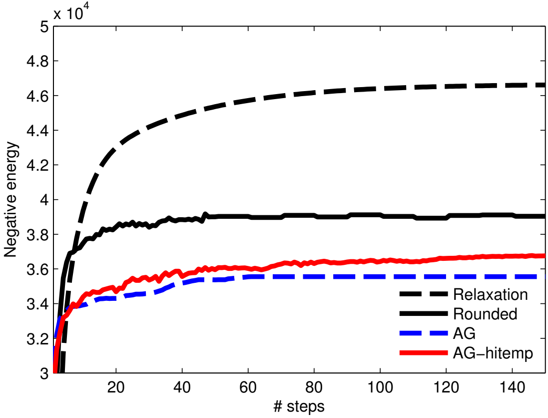


Figure : A comparison of convergence speeds

RRR is fast, and performs well

	seg		dbn	
	time	obj	time	obj
Gurobi	115.68	**3.7958**	301	**2.1013**
A. Gibbs	1.26	3.3537	33	1.9086
$k = 1$	0.15	3.2267	3.7	1.6950
$k = 2$	0.20	3.6830	20	2.0188
$k = 4$	0.49	**3.7653**	12	**2.1119**
$k = 8$	0.86	**3.7643**	5.0	**2.1120**

Table : Averaged results on PASCAL PIC 2011 instances. Time is measured in seconds. DBN is a deep belief networks trained on MNIST

RRR gives us near-MAP samples

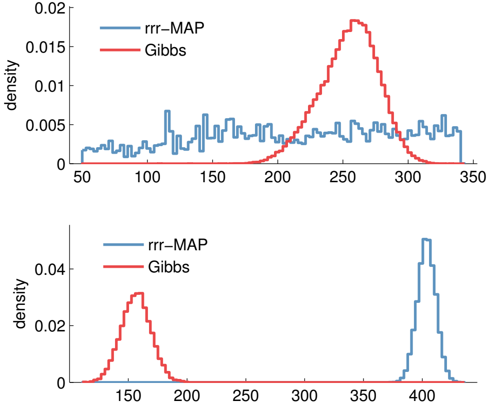
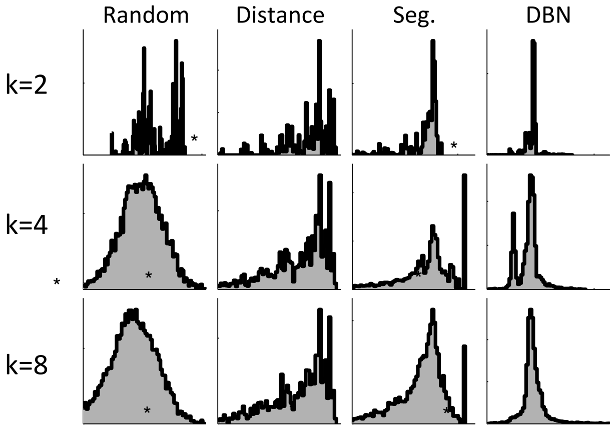


Figure : The negative energy of samples from rrr-MAP compared to Gibbs. top: RBM on MNIST, bot: random matrix

Histograms of the rrr-MAP samples vs. k



Use these samples to estimate the partition function

	True	AIS	rrr-low	rrr-IS
MNIST	-	436.37	436.69	438.40
Random-S	5127.6	5127.5	5095.7	5092.4
Random-L	-	9750.5	9547.7	9606.7

Table : Estimates of the RBM log-partition function $\log Z(A)$

Outline

1. Background
   - The problem
   - Integer quadratic program
2. Randomized Relax and Round
   - Relaxations
   - Randomized rounding
3. Evaluations
4. Conclusions

Current/future works

- Randomized relax-and-round for learning
- Better theoretical bounds specializing to machine learning problems
- Tighten the relaxation using relaxation hierachies

Conclusions

- Proposed the randomized relax-and-round method for MAP inference in MRFs
- Evaluated in the RBB. RRR sometimes performs better than annealed Gibbs, and always give annealed Gibbs a better initialization.
- Generate near-MAP samples