Towards Maximum Likelihood: Learning Undirected Graphical Models using Persistent Sequential Monte Carlo

Hanchen Xiong

Institute of Computer Science
University of Innsbruck, Austria

November 26, 2014
Background

Undirected Graphical Models:

- Markov Random Fields (or Markov Network)
- Conditional Random Fields

Modeling:

\[
p(x; \theta) = \frac{\exp(-E(x; \theta))}{Z(\theta)}
\]  \hspace{1cm} (1)

Energy:

\[
E(x; \theta) = -\theta^\top \phi(x)
\]  \hspace{1cm} (2)

with random variables \( x = [x_1, x_2, \ldots, x_D] \in \mathcal{X}^D \) where \( x_d \) can take \( N_d \) discrete values, \( \phi(x) \) is a \( K \)-dimensional vector of sufficient statistics, and parameter \( \theta \in \mathbb{R}^K \).
Maximum Log-Likelihood Training

- UGMs’ likelihood functions is concave w.r.t. $\theta$
  [Koller and Friedman, 2009];

\[
\begin{align*}
\text{Given training data } D = \{x(m)\}_{m=1}^M, \\
\text{the derivative of average log-likelihood } L(\theta|D) &= \frac{1}{M} \sum_{m=1}^M \log p(x(m); \theta) \\
\frac{\partial L(\theta|D)}{\partial \theta} &= E_D(\phi(x)) \psi - E_{\theta}(\phi(x)) \psi - 1 \sum_{x' \in D} p(x'; \theta) \phi(x) (3)
\end{align*}
\]

interpretation: iteratively pulls down the energy of the data space occupied by $D$ (positive phase), but raises the energy over all data space $X$ (negative phase), until it reaches a balance ($\psi + = \psi -$).
Maximum Log-Likelihood Training

- UGMs’ likelihood functions is concave w.r.t. $\theta$ [Koller and Friedman, 2009];

- Given training data $\mathcal{D} = \{x^{(m)}\}_{m=1}^{M}$, the derivative of average log-likelihood $\mathcal{L}(\theta|\mathcal{D}) = \frac{1}{M}\sum_{m=1}^{M}\log p(x^{(m)}; \theta)$ as

$$\frac{\partial \mathcal{L}(\theta|\mathcal{D})}{\partial \theta} = \mathbb{E}_{\mathcal{D}}(\phi(x)) - \mathbb{E}_{\theta}(\phi(x)) = \frac{1}{M}\sum_{m=1}^{M}\phi(x^{(m)}) - \sum_{x' \in \mathcal{D}}p(x'; \theta)\phi(x)'
$$

(3)
Maximum Log-Likelihood Training

- UGMs’ likelihood functions is concave w.r.t. $\theta$ [Koller and Friedman, 2009];

- Given training data $\mathcal{D} = \{x^{(m)}\}_{m=1}^{M}$, the derivative of average log-likelihood $\mathcal{L}(\theta|\mathcal{D}) = \frac{1}{M} \sum_{m=1}^{M} \log p(x^{(m)}; \theta)$ as

\[
\frac{\partial \mathcal{L}(\theta|\mathcal{D})}{\partial \theta} = \underbrace{\mathbb{E}_\mathcal{D}(\phi(x))}_\psi - \underbrace{\mathbb{E}_\theta(\phi(x))}_\psi = \frac{1}{M} \sum_{m=1}^{M} \phi(x^{(m)}) - \sum_{x' \in \mathcal{D}} p(x'; \theta)\phi(x)'
\]

(3)

- interpretation: iteratively pulls down the energy of the data space occupied by $\mathcal{D}$ (positive phase), but raises the energy over all data space $\mathcal{X}^D$ (negative phase), until it reaches a balance ($\psi^+ = \psi^-$).
Existing Learning Methods

- Approximate the second term of the gradient $\mathcal{L}$:

$$\frac{\partial \mathcal{L}(\theta | \mathcal{D})}{\partial \theta} = \mathbb{E}_D(\phi(x)) - \mathbb{E}_\theta(\phi(x))$$

$\psi^+$ and $\psi^-$
Existing Learning Methods

- Approximate the second term of the gradient $L$:

$$\frac{\partial L(\theta|D)}{\partial \theta} = \mathbb{E}_D(\phi(x)) - \mathbb{E}_\theta(\phi(x))$$

- Markov Chain Monte Carlo Maximum Likelihood (MCMCML) [Geyer, 1991]

- Contrastive Divergence (CD) [Hinton, 2002]

- Persistent Contrastive Divergence (PCD) [Tieleman, 2008]

- Tempered Transition (TT) [Salakhutdinov, 2010]

- Parallel Tempering (PT) [Desjardins et al., 2010]
Existing Learning Methods

- Approximate the second term of the gradient $\mathcal{L}$:

$$\frac{\partial \mathcal{L}(\theta|\mathcal{D})}{\partial \theta} = \mathbb{E}_{\mathcal{D}}(\phi(x)) - \mathbb{E}_{\theta}(\phi(x))$$

- Markov Chain Monte Carlo Maximum Likelihood (MCMCML) [Geyer, 1991]

- Contrastive Divergence (CD) [Hinton, 2002]
Existing Learning Methods

- Approximate the second term of the gradient $\mathcal{L}$:

$$\frac{\partial \mathcal{L}(\theta|\mathcal{D})}{\partial \theta} = \mathbb{E}_\mathcal{D}(\phi(x)) - \mathbb{E}_\theta(\phi(x))$$

- Markov Chain Monte Carlo Maximum Likelihood (MCMCML) [Geyer, 1991]
- Contrastive Divergence (CD) [Hinton, 2002]
- Persistent Contrastive Divergence (PCD) [Tieleman, 2008]
Existing Learning Methods

- Approximate the second term of the gradient $\mathcal{L}$:

$$\frac{\partial \mathcal{L}(\theta | \mathcal{D})}{\partial \theta} = \mathbb{E}_\mathcal{D}(\phi(x)) - \mathbb{E}_\theta(\phi(x))$$

- Markov Chain Monte Carlo Maximum Likelihood (MCMCML) [Geyer, 1991]
- Contrastive Divergence (CD) [Hinton, 2002]
- Persistent Contrastive Divergence (PCD) [Tieleman, 2008]
- Tempered Transition (TT) [Salakhutdinov, 2010]
Existing Learning Methods

- Approximate the second term of the gradient $\mathcal{L}$:
  \[
  \frac{\partial \mathcal{L}(\theta|D)}{\partial \theta} = \underbrace{\mathbb{E}_D(\phi(x))}_{\psi^+} - \underbrace{\mathbb{E}_\theta(\phi(x))}_{\psi^-}
  \]

- Markov Chain Monte Carlo Maximum Likelihood (MCMCML) [Geyer, 1991]
- Contrastive Divergence (CD) [Hinton, 2002]
- Persistent Contrastive Divergence (PCD) [Tieleman, 2008]
- Tempered Transition (TT) [Salakhutdinov, 2010]
- Parallel Tempering (PT) [Desjardins et al., 2010]
Existing Learning Methods

- Approximate the second term of the gradient $\mathcal{L}$:
  \[
  \frac{\partial \mathcal{L}(\theta|\mathcal{D})}{\partial \theta} = \mathbb{E}_\mathcal{D}(\phi(x)) - \mathbb{E}_\theta(\phi(x))
  \]

- Markov Chain Monte Carlo Maximum Likelihood (MCMCML)[Geyer, 1991]
- Contrastive Divergence (CD) [Hinton, 2002]
- Persistent Contrastive Divergence (PCD) [Tieleman, 2008]
- Tempered Transition (TT) [Salakhutdinov, 2010]
- Parallel Tempering (PT) [Desjardins et al., 2010]

- CD, PCD, TT and PT can be summarized as a Robbins-Monro’s stochastic approximation procedure (SAP).
  [Robbins and Monro, 1951]
Robbins-Monro’s SAP

Training data set \( \mathcal{D} = \{ \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)} \} \). Randomly initialize model parameters \( \theta^0 \) and \( N \) particles \( \{ s_{0,1}, \ldots, s_{0,N} \} \).
Robbins-Monro’s SAP

1. Training data set \( \mathcal{D} = \{x^{(1)}, \cdots, x^{(M)}\} \). Randomly initialize model parameters \( \theta^0 \) and \( N \) particles \( \{s^{0,1}, \cdots, s^{0,N}\} \).

2. for \( t = 0 : T \) do // \( T \) iterations

3. Sample \( s^{t+1}, n \) from \( s^t, n \) using transition operator \( H_{\theta^t} \);

4. Update: \( \theta^{t+1} = \theta^t + \eta \left[ \frac{1}{M} \sum_{m=1}^{M} \phi(x^{(m)}) - \frac{1}{N} \sum_{n=1}^{N} \phi(s^{t+1}, n) \right] \)

5. Decrease \( \eta \).

6. end for

When using Gibbs sampler as \( H_{\theta^t} \), SAP becomes PCD, and similarly, \( TT \) and \( PT \) can be substituted as well.
Training data set $\mathcal{D} = \{x^{(1)}, \ldots, x^{(M)}\}$. Randomly initialize model parameters $\theta^0$ and $N$ particles $\{s^{0,1}, \ldots, s^{0,N}\}$.

for $t = 0 : T$ do // T iterations

for $n = 1 : N$ do // go through all N particles

end for

end for

When using Gibbs sampler as $H_{\theta^t}$, SAP becomes PCD, and similarly, TT and PT can be substituted as well.
Robbins-Monro’s SAP

1. Training data set $\mathcal{D} = \{x^{(1)}, \ldots , x^{(M)}\}$. Randomly initialize model parameters $\theta^0$ and $N$ particles $\{s^{0,1}, \ldots , s^{0,N}\}$.

2. for $t = 0 : T$ do // $T$ iterations

3. for $n = 1 : N$ do // go through all $N$ particles

4. Sample $s^{t+1,n}$ from $s^{t,n}$ using transition operator $H_{\theta^t}$;

5. end for

6. end for

8. end for
Robbins-Monro’s SAP

1. Training data set $\mathcal{D} = \{x^{(1)}, \ldots, x^{(M)}\}$. Randomly initialize model parameters $\theta^0$ and $N$ particles $\{s^{0,1}, \ldots, s^{0,N}\}$.

2. for $t = 0 : T$ do // $T$ iterations

3. for $n = 1 : N$ do // go through all $N$ particles

4. Sample $s^{t+1,n}$ from $s^{t,n}$ using transition operator $H_{\theta^t}$;

5. end for

6. Update: $\theta^{t+1} = \theta^t + \eta \left[ \frac{1}{M} \sum_{m=1}^{M} \phi(x^{(m)}) - \frac{1}{N} \sum_{n=1}^{N} \phi(s^{t+1,n}) \right]$

8. end for
Robbins-Monro’s SAP

1. Training data set $\mathcal{D} = \{x^{(1)}, \ldots, x^{(M)}\}$. Randomly initialize model parameters $\theta^0$ and $N$ particles $\{s^{0,1}, \ldots, s^{0,N}\}$.

2. for $t = 0 : T$ do // $T$ iterations

3. for $n = 1 : N$ do // go through all N particles

4. Sample $s^{t+1,n}$ from $s^{t,n}$ using transition operator $H_{\theta^t}$;

5. end for

6. Update: $\theta^{t+1} = \theta^t + \eta \left[ \frac{1}{M} \sum_{m=1}^{M} \phi(x^{(m)}) - \frac{1}{N} \sum_{n=1}^{N} \phi(s^{t+1,n}) \right]$.

7. Decrease $\eta$.

8. end for
Robbins-Monro’s SAP

1. Training data set $D = \{x^{(1)}, \ldots, x^{(M)}\}$. Randomly initialize model parameters $\theta^0$ and $N$ particles $\{s^{0,1}, \ldots, s^{0,N}\}$.

2. for $t = 0 : T$ do // T iterations
3.   for $n = 1 : N$ do // go through all N particles
4.     Sample $s^{t+1,n}$ from $s^{t,n}$ using transition operator $H_{\theta^t}$;
5.   end for
6. Update: $\theta^{t+1} = \theta^t + \eta \left[ \frac{1}{M} \sum_{m=1}^{M} \phi(x^{(m)}) - \frac{1}{N} \sum_{n=1}^{N} \phi(s^{t+1,n}) \right]$
7. Decrease $\eta$.
8. end for

9. When using Gibbs sampler as $H_{\theta^t}$, SAP becomes PCD, and similarly, TT and PT can be substituted as well.
In MCMCML, a proposal distribution \( p(x; \theta_0) \) is set up in the same form as a UGM, and we have

\[
\frac{Z(\theta)}{Z(\theta_0)} = \frac{\sum_x \exp(\theta^T \phi(x))}{\sum_x \exp(\theta_0^T \phi(x))} \times \frac{\exp(\theta_0^T \phi(x))}{\sum_x \exp(\theta_0^T \phi(x))}
\]

\[
= \sum_x \frac{\exp(\theta^T \phi(x))}{\exp(\theta_0^T \phi(x))} \times \frac{\exp(\theta_0^T \phi(x))}{\sum_x \exp(\theta_0^T \phi(x))}
\]

\[
= \sum_x \exp \left( (\theta - \theta_0)^T \phi(x) \right) p(x; \theta_0)
\]

\[
\approx \frac{1}{S} \sum_{s=1}^{S} w^{(s)}
\]

where \( w^{(s)} \) is

\[
w^{(s)} = \exp \left( (\theta - \theta_0)^T \phi(\tilde{x}^{(s)}) \right),
\]
In MCMCML, a proposal distribution $p(x; \theta_0)$ is set up in the same form as a UGM, and we have

$$\frac{Z(\theta)}{Z(\theta_0)} = \frac{\sum_x \exp(\theta^T \phi(x))}{\sum_x \exp(\theta_0^T \phi(x))} \times \frac{\exp(\theta_0^T \phi(x))}{\sum_x \exp(\theta_0^T \phi(x))}$$

$$= \frac{\sum_x \exp(\theta^T \phi(x))}{\sum_x \exp(\theta_0^T \phi(x))} \times \sum_x \exp(\theta_0^T \phi(x))$$

$$= \sum_x \exp\left(\left(\theta - \theta_0\right)^T \phi(x)\right) p(x; \theta_0)$$

$$\approx \frac{1}{S} \sum_{s=1}^S w^{(s)}$$

where $w^{(s)}$ is

$$w^{(s)} = \exp\left(\left(\theta - \theta_0\right)^T \phi(\bar{x}^{(s)})\right),$$

MCMCML is an importance sampling approximation of the gradient.
MCMCML

1. $t \leftarrow 0$, initialize the proposal distribution $p(x; \theta_0)$
2. Sample $\{\bar{x}^{(s)}\}$ from $p(x; \theta_0)$
3. while ! stop criterion do
   4. Calculate $w^{(s)}$ using (5)
   5. Calculate gradient $\frac{\partial \tilde{L}(\theta_t|\mathcal{D})}{\partial \theta_t}$ using importance sampling approximation.
   6. update $\theta_{t+1} = \theta_t + \eta \frac{\partial \tilde{L}(\theta_t|\mathcal{D})}{\partial \theta_t}$
   7. $t \leftarrow t + 1$
4. end while
MCMCML's performance highly depends on initial proposal distribution;
MCMCML’s performance highly depends on initial proposal distribution;

at time $t$, it is helpful to update the proposal distribution as the $p(x; \theta_{t-1})$;
MCMCML

- MCMCML's performance highly depends on initial proposal distribution;
- at time $t$, it is helpful to update the proposal distribution as the $p(x; \theta_{t-1})$;
- this is analogous to sequential importance sampling with resampling at every iteration, however, the construction of sequential distributions is by learning;
MCMCML

- MCMCML’s performance highly depends on initial proposal distribution;
- at time $t$, it is helpful to update the proposal distribution as the $p(x; \theta_{t-1})$;
- this is analogous to **sequential importance sampling** with resampling at every iteration, however, the construction of sequential distributions is by learning;
- this also looks like SAP learning schemes.
MCMCML’s performance highly depends on initial proposal distribution;

at time $t$, it is helpful to update the proposal distribution as the $p(x; \theta_{t-1})$;

this is analogous to sequential importance sampling with resampling at every iteration, however, the construction of sequential distributions is by learning;

this also looks like SAP learning schemes.

a similar connection between PCD and Sequential Monte Carlo was found in [Asuncion et al., 2010]
SAP Learning as Sequential Monte Carlo

1. Initialize $p(x; \theta_0)$, $t \leftarrow 0$
2. Sample particles $\{\bar{x}_0^{(s)}\}_{s=1}^S \sim p(x; \theta_0)$
3. while ! stop criterion do
4. Assign $w^{(s)} \leftarrow \frac{1}{S}, \forall s \in S$ // importance reweighting
5. // resampling is ignored because it has no effect
6. switch (algorithmic choice) // MCMC transition
7. case CD:
8. generate a brand new particle set $\{\bar{x}_{t+1}^{(s)}\}_{s=1}^S$ with Gibbs sampling from $D$
9. case PCD:
10. evolve particle set $\{\bar{x}_t^{(s)}\}_{s=1}^S$ to $\{\bar{x}_{t+1}^{(s)}\}_{s=1}^S$ with one step Gibbs sampling
11. case Tempered Transition:
12. evolve particle set $\{\bar{x}_t^{(s)}\}_{s=1}^S$ to $\{\bar{x}_{t+1}^{(s)}\}_{s=1}^S$ with tempered transition
13. case Parallel Tempering:
14. evolve particle set $\{\bar{x}_t^{(s)}\}_{s=1}^S$ to $\{\bar{x}_{t+1}^{(s)}\}_{s=1}^S$ with parallel tempering
15. end switch
16. Compute the gradient $\Delta \theta_t$ according to (4);
17. $\theta_{t+1} = \theta_t + \eta \Delta \theta_t$, $t \leftarrow t + 1$;
18. reduce $\eta$;
19. end while
A sequential Monte Carlo (SMC) algorithms can work on the condition that sequential, intermediate distributions are well constructed: two successive ones should be close.
A sequential Monte Carlo (SMC) algorithms can work on the condition that sequential, intermediate distributions are well constructed: two successive ones should be close.

the gap between successive distributions in SAP: $\eta \times D$

1. learning rate $\eta$;
2. the dimensionality of $x$: $D$. 
Persistent Sequential Monte Carlo

- Every sequential, intermediate distributions is constructed by learning, so learning and sampling are interestingly entangled.
Every sequential, intermediate distributions is constructed by learning, so learning and sampling are interestingly entangled. applying SMC philosophy future in sampling: Persistent SMC (SPMC)
Persistent Sequential Monte Carlo

\[
\{\bar{x}^{(s)}\}_{s=1}^{S/2} \sim \mathcal{U}^D
\]

\[
\{\bar{x}^{t-1,(s)}\}_{s=1}^{S/2} \sim p_H(x; \theta_{t-1})
\]

\[
p_0(x; \theta_t) \rightarrow p_1(x; \theta_t) \rightarrow \cdots \rightarrow p_H(x; \theta_t)
\]

\[
U^D \text{ is a uniform distribution on } x, \text{ and intermediate sequential distributions are: } p_h(x; \theta_{t+1}) \propto p(x; \theta_t)^{1-\beta_h} p(x; \theta_{t+1})^{\beta_h}
\]

where \(0 \leq \beta_H \leq \beta_{H-1} \leq \cdots \beta_0 = 1\).
One issue arising in PSMC is the number of $\beta_h$, i.e. $H$.

where $\Delta \beta_h$ is the step length from $\beta_h - 1$ to $\beta_h$, i.e. $\Delta \beta_h = \beta_h - \beta_{h-1}$.
One issue arising in PSMC is the number of $\beta_h$, i.e. $H$

By exploiting degeneration of particle set: the importance weighting for each particle is

$$w^{(s)} = \frac{p_h(\bar{x}^{(s)}; \theta_{t+1})}{p_{h-1}(\bar{x}^{(s)}; \theta_{t+1})} = \exp\left(E(\bar{x}^{(s)}; \theta_t)\Delta\beta_h\right) \exp\left(E(\bar{x}^{(s)}; \theta_{t+1})\right)^{-\Delta\beta_h}$$

(6)

where $\Delta\beta_h$ is the step length from $\beta_{h-1}$ to $\beta_h$, i.e. $\Delta\beta_h = \beta_h - \beta_{h-1}$.

The ESS of a particle set as [Kong et al., 1994]

$$\sigma = \frac{\left(\sum_{s=1}^{S} w^{(s)}\right)^2}{S \sum_{s=1}^{S} w^{(s)}^2} \in \left[\frac{1}{S}, 1\right]$$

(7)
One issue arising in PSMC is the number of $\beta_h$, i.e. $H$

By exploiting degeneration of particle set: the importance weighting
for each particle is

$$w^{(s)} = \frac{p_h(\bar{x}^{(s)}; \theta_{t+1})}{p_{h-1}(\bar{x}^{(s)}; \theta_{t+1})}$$

$$= \exp \left( E(\bar{x}^{(s)}; \theta_{t}) \right)^{\Delta \beta_h} \exp \left( E(\bar{x}^{(s)}; \theta_{t+1}) \right)^{-\Delta \beta_h}$$

(6)

where $\Delta \beta_h$ is the step length from $\beta_{h-1}$ to $\beta_h$, i.e. $\Delta \beta_h = \beta_h - \beta_{h-1}$.

the ESS of a particle set as [Kong et al., 1994]

$$\sigma = \left( \frac{\sum_{s=1}^{S} w^{(s)}}{S} \right)^2 \in \left[ \frac{1}{S}, 1 \right]$$

(7)

ESS $\sigma$ is actually a function of $\Delta \beta_h$. 
Number of Sub-Sequential Distributions

- One issue arising in PSMC is the number of $\beta_h$, i.e. $H$
- By exploiting degeneration of particle set: the importance weighting for each particle is

$$w^{(s)} = \frac{p_h(\bar{x}^{(s)}; \theta_{t+1})}{p_{h-1}(\bar{x}^{(s)}; \theta_{t+1})}$$

$$= \exp\left(E(\bar{x}^{(s)}; \theta_t)\right)^{\Delta \beta_h} \exp\left(E(\bar{x}^{(s)}; \theta_{t+1})\right)^{-\Delta \beta_h}$$

where $\Delta \beta_h$ is the step length from $\beta_{h-1}$ to $\beta_h$, i.e. $\Delta \beta_h = \beta_h - \beta_{h-1}$.

- The ESS of a particle set as [Kong et al., 1994]

$$\sigma = \frac{\left(\sum_{s=1}^{S} w^{(s)}\right)^2}{S \sum_{s=1}^{S} w^{(s)^2}} \in \left[\frac{1}{S}, 1\right]$$

- ESS $\sigma$ is actually a function of $\Delta \beta_h$.
- Set a threshold on $\sigma$, at every $h$, and find the biggest gap by using bidirectional search.
PSMC

**Input:** a training dataset $\mathcal{D} = \{\mathbf{x}^{(m)}\}_{m=1}^{M}$, learning rate $\eta$

1. Initialize $p(\mathbf{x}; \theta_0)$, $t \leftarrow 0$
2. Sample particles $\{\bar{\mathbf{x}}_0^{(s)}\}_{s=1}^{S} \sim p(\mathbf{x}; \theta_0)$
3. while ! stop criterion // root-SMC do
   4. $h \leftarrow 0$, $\beta_0 \leftarrow 1$
   5. while $\beta_h < 1$ // sub-SMC do
      6. assign importance weights $\{w^{(s)}\}_{s=1}^{S}$ to particles according to (6)
      7. resample particles based on $\{w^{(s)}\}_{s=1}^{S}$
      8. find the step length $\Delta \beta_h$
      9. $\beta_{h+1} = \beta_h + \delta \beta$
     10. $h \leftarrow h + 1$
   11. end while
   12. Compute the gradient $\Delta \theta_t$ according to (4)
13. $\theta_{t+1} = \theta_t + \eta \Delta \theta_t$
14. $t \leftarrow t + 1$
15. end while
Experiments

two experiments on two challenges:

- big learning rates
- high dimensional distributions
First Experiment: Small Learning Rates

A small-size Boltzmann Machine with only 10 variables is used to avoid the effect of model complexity. The learning rate $\eta_t = \frac{1}{100+t}$.

Figure: The performance of algorithms with the first learning rate scheme. (a): log-likelihood vs. number of epochs; (b) and (c): the number of $\beta$'s in PSMC and SMC at each iteration (blue) and their mean values (red).
First Experiment: Bigger Learning Rates

learning rate $\eta_t = \frac{1}{20 + 0.5 \times t}$

Figure: The performance of algorithms with the second learning rate scheme. (a): log-likelihood vs. number of epochs; (b) and (c): the number of $\beta$s in PSMC and SMC at each iteration (blue) and their mean values (red).
First Experiment: Large Learning Rates

Learning rate $\eta_t = \frac{1}{10 + 0.1 \times t}$

Figure: The performance of algorithms with the third learning rate scheme. (a): log-likelihood vs. number of epochs; (b) and (c): the number of $\beta$s in PSMC and SMC at each iteration (blue) and their mean values (red).
Second Experiment: Small Dimensionality/Scale

we used the popular restricted Boltzmann machine to model handwritten
digit images (the MNIST database).

10 hidden unites
Second Experiment: Large Dimensionality/Scale

500 hidden units

(d)

(e)

(f)

Hanchen Xiong (UIBK) Learning UGMs with Persistent SMC November 26, 2014 20 / 22
a new interpretation of learning undirected graphical models: sequential Monte Carlo (SMC)
Conclusion

- a new interpretation of learning undirected graphical models: sequential Monte Carlo (SMC)
- reveal two challenges in learning: large learning rate, high dimensionality;
Conclusion

- a new interpretation of learning undirected graphical models: sequential Monte Carlo (SMC)
- reveal two challenges in learning: large learning rate, high dimensionality;
- deeper application of SMC in learning → Persistent SMC;
a new interpretation of learning undirected graphical models: sequential Monte Carlo (SMC)
reveal two challenges in learning: large learning rate, high dimensionality;
deeper application of SMC in learning → Persistent SMC;
yield higher likelihood than state-of-the-art algorithms in challenging cases.
END


*Probabilistic Graphical Models: Principles and Techniques.*  
MIT Press.

Sequential Imputations and Bayesian Missing Data Problems.  

A Stochastic Approximation Method.  

Salakhutdinov, R. (2010).  
Learning in markov random fields using tempered transitions.  
In *NIPS*.

Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient.  
In *ICML*, pages 1064–1071.