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)
- Condtional Random Fields

Modeling:

$$p(\mathbf{x};\boldsymbol{\theta}) = \frac{\exp\left(-E(\mathbf{x};\boldsymbol{\theta})\right)}{\mathbf{Z}(\boldsymbol{\theta})} \quad (1)$$

$$\text{Energy:} \quad E(\mathbf{x};\boldsymbol{\theta}) = -\boldsymbol{\theta}^\top \phi(\mathbf{x}) \quad (2)$$

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

Maximum Log-Likelihood Training

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

Maximum Log-Likelihood Training

- UGMs' likelihood functions is concave w.r.t. $\boldsymbol{\theta}$ [Koller and Friedman, 2009];
- Given training data $\mathcal{D} = \{\mathbf{x}^{(m)}\}_{m=1}^{M}$, the derivative of average log-likelihood $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{M}\sum_{m=1}^{M} \log p(\mathbf{x}^{(m)};\boldsymbol{\theta})$ as

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^+} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^-} = \frac{1}{M}\sum_{m=1}^{M}\phi(\mathbf{x}^{(m)}) - \sum_{\mathbf{x}'\in\mathcal{D}} p(\mathbf{x}';\boldsymbol{\theta})\phi(\mathbf{x})' \tag{3}$$

Maximum Log-Likelihood Training

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

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

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^+} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^-} = \frac{1}{M}\sum_{m=1}^{M} \phi(\mathbf{x}^{(m)}) - \sum_{\mathbf{x}' \in \mathcal{D}} p(\mathbf{x}'; \boldsymbol{\theta})\phi(\mathbf{x})' \tag{3}$$

- 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 $\mathcal{L}$:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^{+}} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^{-}}$$

Existing Learning Methods

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

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^{+}} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^{-}}$$

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

Existing Learning Methods

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

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^{+}} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^{-}}$$

- Markov Chain Monte Carlo Maximum Likelihood (MCMCML)[Geyer, 1991]
- Contrastive Divergence (CD) [Hinton, 2002]

Existing Learning Methods

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

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^+} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^-}$$

- 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 $\mathcal{L}$:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^+} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^-}$$

- 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]

Existing Learning Methods

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

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^{+}} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^{-}}$$

- 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 $\mathcal{L}$:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathcal{D})}{\partial \boldsymbol{\theta}} = \underbrace{\mathbb{E}_{\mathcal{D}}(\phi(\mathbf{x}))}_{\psi^+} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}}(\phi(\mathbf{x}))}_{\psi^-}$$

- 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]
- 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

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

Robbins-Monro's SAP

1. Training data set $\mathcal{D} = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}\}$. Randomly initialize model parameters $\boldsymbol{\theta}^0$ and N particles $\{\mathbf{s}^{0,1}, \cdots, \mathbf{s}^{0,N}\}$.
2. **for** $t = 0 : T$ **do** // T iterations

8. **end for**

Robbins-Monro's SAP

1. Training data set $\mathcal{D} = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}\}$. Randomly initialize model parameters θ^0 and N particles $\{\mathbf{s}^{0,1}, \cdots, \mathbf{s}^{0,N}\}$.
2. **for** $t = 0 : T$ **do** // T iterations
3. **for** $n = 1 : N$ **do** // go through all N particles
5. **end for**
8. **end for**

Robbins-Monro's SAP

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

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

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

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

  end for

end for

Robbins-Monro's SAP

1. Training data set $\mathcal{D} = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}\}$. Randomly initialize model parameters $\boldsymbol{\theta}^0$ and N particles $\{\mathbf{s}^{0,1}, \cdots, \mathbf{s}^{0,N}\}$.
2. **for** $t = 0 : T$ **do** // T iterations
3. **for** $n = 1 : N$ **do** // go through all N particles
4. Sample $\mathbf{s}^{t+1,n}$ from $\mathbf{s}^{t,n}$ using transition operator $H_{\boldsymbol{\theta}^t}$;
5. **end for**
6. Update: $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \eta\Big[\frac{1}{M}\sum_{m=1}^{M}\phi(\mathbf{x}^{(m)}) - \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{s}^{t+1,n})\Big]$
8. **end for**

Robbins-Monro's SAP

1. Training data set $\mathcal{D} = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}\}$. Randomly initialize model parameters $\boldsymbol{\theta}^0$ and N particles $\{\mathbf{s}^{0,1}, \cdots, \mathbf{s}^{0,N}\}$.
2. **for** $t = 0 : T$ **do** // T iterations
3. **for** $n = 1 : N$ **do** // go through all N particles
4. Sample $\mathbf{s}^{t+1,n}$ from $\mathbf{s}^{t,n}$ using transition operator $H_{\boldsymbol{\theta}^t}$;
5. **end for**
6. Update: $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \eta\Big[\frac{1}{M}\sum_{m=1}^{M}\phi(\mathbf{x}^{(m)}) - \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{s}^{t+1,n})\Big]$
7. Decrease η.
8. **end for**

Robbins-Monro's SAP

1. Training data set $\mathcal{D} = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}\}$. Randomly initialize model parameters $\boldsymbol{\theta}^0$ and N particles $\{\mathbf{s}^{0,1}, \cdots, \mathbf{s}^{0,N}\}$.
2. **for** $t = 0 : T$ **do** // T iterations
3. **for** $n = 1 : N$ **do** // go through all N particles
4. Sample $\mathbf{s}^{t+1,n}$ from $\mathbf{s}^{t,n}$ using transition operator $H_{\boldsymbol{\theta}^t}$;
5. **end for**
6. Update: $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \eta\Big[\frac{1}{M}\sum_{m=1}^{M}\phi(\mathbf{x}^{(m)}) - \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{s}^{t+1,n})\Big]$
7. Decrease η.
8. **end for**
9. When using Gibbs sampler as $H_{\boldsymbol{\theta}^t}$, SAP becomes PCD, and similarly, TT and PT can be substituted as well.

MCMCML

- In MCMCML, a proposal distribution $p(\mathbf{x};\boldsymbol{\theta_0})$ is set up in the same form as a UGM, and we have

$$\begin{aligned}
\frac{\mathbf{Z}(\boldsymbol{\theta})}{\mathbf{Z}(\boldsymbol{\theta}_0)} &= \frac{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}^\top \phi(\mathbf{x}))}{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \\
&= \frac{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}^\top \phi(\mathbf{x}))}{\exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \times \frac{\exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))}{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \\
&= \sum_{\mathbf{x}} \frac{\exp(\boldsymbol{\theta}^\top \phi(\mathbf{x}))}{\exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \times \frac{\exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))}{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \\
&= \sum_{\mathbf{x}} \exp\left((\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \phi(\mathbf{x})\right) p(\mathbf{x};\boldsymbol{\theta}_0) \\
&\approx \frac{1}{S} \sum_{s=1}^{S} w^{(s)}
\end{aligned} \tag{4}$$

where $w^{(s)}$ is

$$w^{(s)} = \exp\left((\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \phi(\bar{\mathbf{x}}^{(s)})\right), \tag{5}$$

MCMCML

- In MCMCML, a proposal distribution $p(\mathbf{x}; \boldsymbol{\theta_0})$ is set up in the same form as a UGM, and we have

$$
\begin{aligned}
\frac{\mathbf{Z}(\boldsymbol{\theta})}{\mathbf{Z}(\boldsymbol{\theta}_0)} &= \frac{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}^\top \phi(\mathbf{x}))}{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \\
&= \frac{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}^\top \phi(\mathbf{x}))}{\exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \times \frac{\exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))}{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \\
&= \sum_{\mathbf{x}} \frac{\exp(\boldsymbol{\theta}^\top \phi(\mathbf{x}))}{\exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \times \frac{\exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))}{\sum_{\mathbf{x}} \exp(\boldsymbol{\theta}_0^\top \phi(\mathbf{x}))} \\
&= \sum_{\mathbf{x}} \exp\left((\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \phi(\mathbf{x})\right) p(\mathbf{x}; \boldsymbol{\theta}_0) \\
&\approx \frac{1}{S} \sum_{s=1}^{S} w^{(s)}
\end{aligned}
\tag{4}
$$

where $w^{(s)}$ is

$$
w^{(s)} = \exp\left((\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \phi(\bar{\mathbf{x}}^{(s)})\right), \tag{5}
$$

- MCMCML is an importance sampling approximation of the gradient.

MCMCML

$t \leftarrow 0$, initialize the proposal distribution $p(\mathbf{x}; \boldsymbol{\theta}_0)$
Sample $\{\bar{\mathbf{x}}^{(s)}\}$ from $p(\mathbf{x}; \boldsymbol{\theta}_0)$
while ! stop criterion **do**
    Calculate $w^{(s)}$ using (5)
    Calculate gradient $\frac{\partial \tilde{\mathcal{L}}(\boldsymbol{\theta}_t|\mathcal{D})}{\partial \boldsymbol{\theta}_t}$ using importance sampling approximation.
    update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta \frac{\partial \tilde{\mathcal{L}}(\boldsymbol{\theta}_t|\mathcal{D})}{\partial \boldsymbol{\theta}_t}$
    $t \leftarrow t + 1$
end while

MCMCML

- MCMCML's performance highly depends on initial proposal distribution;

MCMCML

- MCMCML's performance highly depends on initial proposal distribution;
- at time t, it is helpful to update the proposal distribution as the $p(\mathbf{x}; \boldsymbol{\theta}_{t-1})$;

MCMCML

- MCMCML's performance highly depends on initial proposal distribution;
- at time t, it is helpful to update the proposal distribution as the $p(\mathbf{x}; \boldsymbol{\theta}_{t-1})$;
- 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 $p(\mathbf{x}; \boldsymbol{\theta}_{t-1})$;
- 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

- MCMCML's performance highly depends on initial proposal distribution;
- at time t, it is helpful to update the proposal distribution as the $p(\mathbf{x}; \boldsymbol{\theta}_{t-1})$;
- 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

Initialize $p(\mathbf{x}; \boldsymbol{\theta}_0)$, $t \leftarrow 0$
Sample particles $\{\bar{\mathbf{x}}_0^{(s)}\}_{s=1}^{S} \sim p(\mathbf{x}; \boldsymbol{\theta}_0)$
while ! stop criterion **do**
  Assign $w^{(s)} \leftarrow \frac{1}{S}, \forall s \in S$ `// importance reweighting`
  `// resampling is ignored because it has no effect`
  switch (algorithmic choice) `// MCMC transition`
  case CD:
    generate a brand new particle set $\{\bar{\mathbf{x}}_{t+1}^{(s)}\}_{s=1}^{S}$ with Gibbs sampling from $\mathcal{D}$
  case PCD:
    evolve particle set $\{\bar{\mathbf{x}}_t^{(s)}\}_{s=1}^{S}$ to $\{\bar{\mathbf{x}}_{t+1}^{(s)}\}_{s=1}^{S}$ with one step Gibbs sampling
  case Tempered Transition:
    evolve particle set $\{\bar{\mathbf{x}}_t^{(s)}\}_{s=1}^{S}$ to $\{\bar{\mathbf{x}}_{t+1}^{(s)}\}_{s=1}^{S}$ with tempered transition
  case Parallel Tempering:
    evolve particle set $\{\bar{\mathbf{x}}_t^{(s)}\}_{s=1}^{S}$ to $\{\bar{\mathbf{x}}_{t+1}^{(s)}\}_{s=1}^{S}$ with parallel tempering
  end switch
  Compute the gradient $\Delta\boldsymbol{\theta}_t$ according to (4);
  $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta\Delta\boldsymbol{\theta}_t$, $t \leftarrow t+1$;
  reduce η;
end while

SAP Learning as Sequential Monte Carlo

- A sequential Monte Carlo (SMC) algorithms can work on the condition that sequential, intermediate distributions are well constructed: two successive ones should be close.

SAP Learning as Sequential Monte Carlo

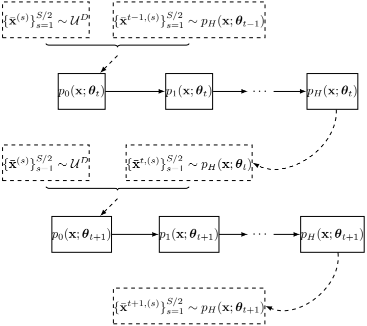
- A sequential Monte Carlo (SMC) algorithms can work on the condition that sequential, intermediate distributions are well constructed: two successive ones should be close.
- the gap between successive distributions in SAP: $\eta \times D$
  1. learning rate η;
  2. the dimensionality of $\mathbf{x}$: D.

Persistent Sequential Monte Carlo

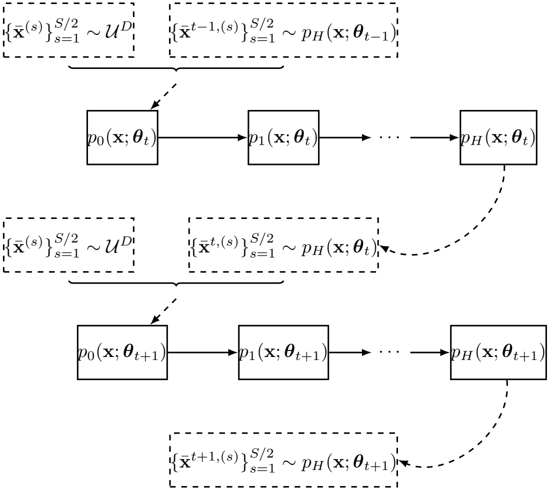
- Every sequential, intermediate distributions is constructed by learning, so learning and sampling are interestingly entangled.

Persistent Sequential Monte Carlo

- 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



$\mathcal{U}^D$ is a uniform distribution on $\mathbf{x}$, and intermediate sequential distributions are: $p_h(\mathbf{x};\boldsymbol{\theta}_{t+1}) \propto p(\mathbf{x};\boldsymbol{\theta}_t)^{1-\beta_h} p(\mathbf{x};\boldsymbol{\theta}_{t+1})^{\beta_h}$ where $0 \leq \beta_H \leq \beta_{H-1} \leq \cdots \beta_0 = 1$.

Number of Sub-Sequential Distributions

- One issue arising in PSMC is the number of β_h, *i.e.* H

Number of Sub-Sequential Distributions

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

$$\begin{aligned} w^{(s)} &= \frac{p_h(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})}{p_{h-1}(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})} \\ &= \exp\left(E(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_t)\right)^{\Delta\beta_h} \exp\left(E(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})\right)^{-\Delta\beta_h} \end{aligned} \tag{6}$$

where $\Delta\beta_h$ is the step length from β_{h-1} to β_h, *i.e.* $\Delta\beta_h = \beta_h - \beta_{h-1}$. the ESS of a particle set as [Kong et al., 1994]

$$\sigma = \frac{(\sum_{s=1}^{S} w^{(s)})^2}{S \sum_{s=1}^{S} w^{(s)2}} \in \left[\frac{1}{S}, 1\right] \tag{7}$$

Number of Sub-Sequential Distributions

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

$$w^{(s)} = \frac{p_h(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})}{p_{h-1}(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})} = \exp\left(E(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_t)\right)^{\Delta\beta_h} \exp\left(E(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})\right)^{-\Delta\beta_h} \tag{6}$$

where $\Delta\beta_h$ is the step length from β_{h-1} to β_h, *i.e.* $\Delta\beta_h = \beta_h - \beta_{h-1}$. the ESS of a particle set as [Kong et al., 1994]

$$\sigma = \frac{(\sum_{s=1}^{S} w^{(s)})^2}{S \sum_{s=1}^{S} w^{(s)2}} \in \left[\frac{1}{S}, 1\right] \tag{7}$$

- ESS σ is actually a function of $\Delta\beta_h$.

Number of Sub-Sequential Distributions

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

$$\begin{aligned} w^{(s)} &= \frac{p_h(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})}{p_{h-1}(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})} \\ &= \exp\left(E(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_t)\right)^{\Delta\beta_h} \exp\left(E(\bar{\mathbf{x}}^{(s)}; \boldsymbol{\theta}_{t+1})\right)^{-\Delta\beta_h} \end{aligned} \tag{6}$$

where $\Delta\beta_h$ is the step length from β_{h-1} to β_h, *i.e.* $\Delta\beta_h = \beta_h - \beta_{h-1}$. the ESS of a particle set as [Kong et al., 1994]

$$\sigma = \frac{\left(\sum_{s=1}^{S} w^{(s)}\right)^2}{S \sum_{s=1}^{S} w^{(s)2}} \in \left[\frac{1}{S}, 1\right] \tag{7}$$

- ESS σ is actually a function of $\Delta\beta_h$.
- Set a threshold on σ, 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 η

Initialize $p(\mathbf{x}; \boldsymbol{\theta}_0)$, $t \leftarrow 0$
Sample particles $\{\bar{\mathbf{x}}_0^{(s)}\}_{s=1}^{S} \sim p(\mathbf{x}; \boldsymbol{\theta}_0)$
while ! stop criterion `// root-SMC` **do**
    $h \leftarrow 0$, $\beta_0 \leftarrow 1$
    while $\beta_h < 1$ `// sub-SMC` **do**
        assign importance weights $\{w^{(s)}\}_{s=1}^{S}$ to particles according to (6)
        resample particles based on $\{w^{(s)}\}_{s=1}^{S}$
        find the step length $\Delta\beta_h$
        $\beta_{h+1} = \beta_h + \delta\beta$
        $h \leftarrow h + 1$
    end while
    Compute the gradient $\Delta\boldsymbol{\theta}_t$ according to (4)
    $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta\Delta\boldsymbol{\theta}_t$
    $t \leftarrow t + 1$
end while

Experiments

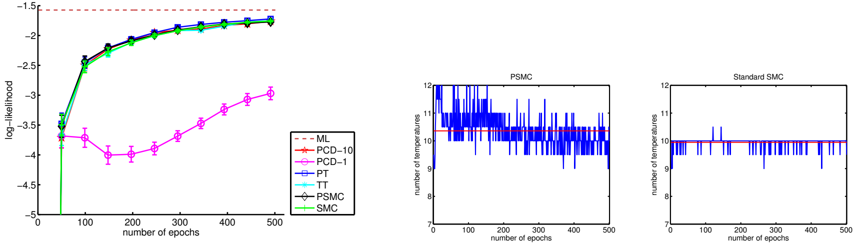
two experiments on two challenges:

- 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

learning rate $\eta_t = \frac{1}{100+t}$

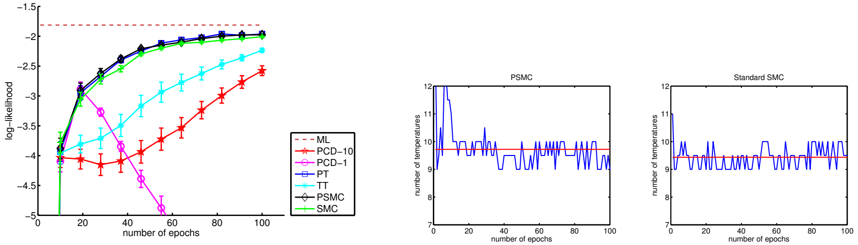


(a) (b) (c)

Figure: The performance of algorithms with the first learning rate scheme. (a): log-likelihood *vs.* number of epochs; (b) and (c): the number of βs in PSMC and SMC at each iteration (blue) and their mean values (red).

First Experiment: Bigger Learning Rates

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

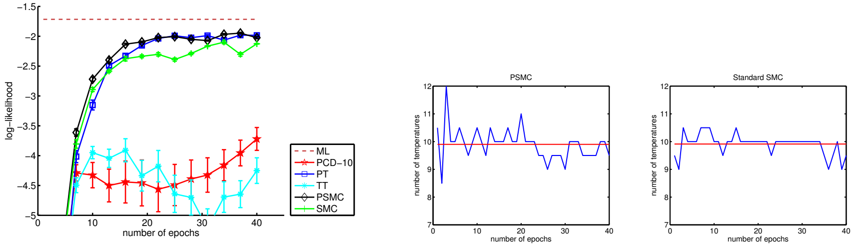


(a) (b) (c)

Figure: The performance of algorithms with the second learning rate scheme. (a): log-likelihood *vs.* number of epochs; (b) and (c): the number of βs in PSMC and SMC at each iteration (blue) and their mean values (red).

First Experiment: Large Learning Rates

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



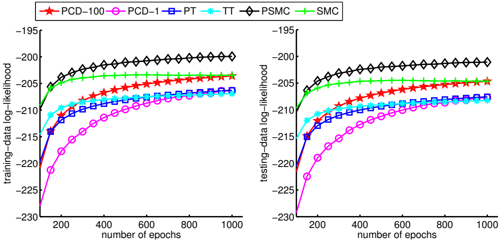
(a) (b) (c)

Figure: The performance of algorithms with the third learning rate scheme. (a): log-likelihood *vs.* number of epochs; (b) and (c): the number of βs in PSMC and SMC at each iteration (blue) and their mean values (red).

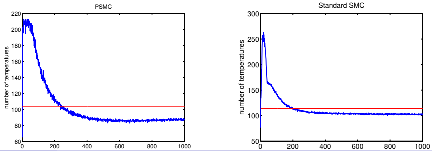
Second Experiment: Small Dimensionality/Scale

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

10 hidden unites

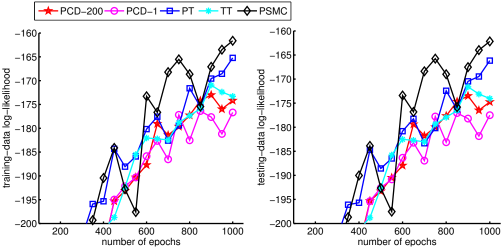


(a)

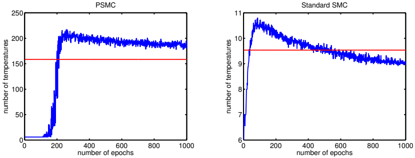


Second Experiment: Large Dimensionality/Scale

500 hidden unites



(d)



(e) (f)

Conclusion

- 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 $\longrightarrow$ Persistent SMC;

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 $\longrightarrow$ Persistent SMC;
- yield higher likelihood than state-of-the-art algorithms in challenging cases.

END

Asuncion, A. U., Liu, Q., Ihler, A. T., and Smyth, P. (2010).
Particle filtered MCMC-MLE with connections to contrastive divergence.
In *ICML*.

Desjardins, G., Courville, A., Bengio, Y., Vincent, P., and Delalleau, O. (2010).
Tempered Markov Chain Monte Carlo for training of restricted Boltzmann machines.
In *AISTATS*.

Geyer, C. J. (1991).
Markov Chain Monte Carlo Maximum Likelihood.
Compuuting Science and Statistics:Proceedings of the 23rd Symposium on the Interface.

Hinton, G. E. (2002).
Training products of experts by minimizing contrastive divergence.
Neural Computation, 14(8):1771–1800.

Koller, D. and Friedman, N. (2009).
Probabilistic Graphical Models: Principles and Techniques.
MIT Press.

Kong, A., Liu, J. S., and Wong, W. H. (1994).
Sequential Imputations and Bayesian Missing Data Problems.
Journal of the American Statistical Association, 89(425):278–288.

Robbins, H. and Monro, S. (1951).
A Stochastic Approximation Method.
Ann.Math.Stat., 22:400–407.

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

Tieleman, T. (2008).
Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient.
In *ICML*, pages 1064–1071.