On Tracking The Partition Function: Supplemental Material

Guillaume Desjardins, Aaron Courville, Yoshua Bengio
{desjagui,courvila,bengioy}@iro.umontreal.ca
Département d'informatique et de recherche opérationnelle
Université de Montréal

The supplementary material provides a more comprehensive view of our log partition function tracking algorithm. The notation is mostly identical to that of the paper and is summarized below for convenience. One minor difference however, is that we make widespread use of the notations $A_{:,l}$ and $A_{m,:}$ to indicate the vectors formed by the l-th column and m-th row of matrix A (respectively).

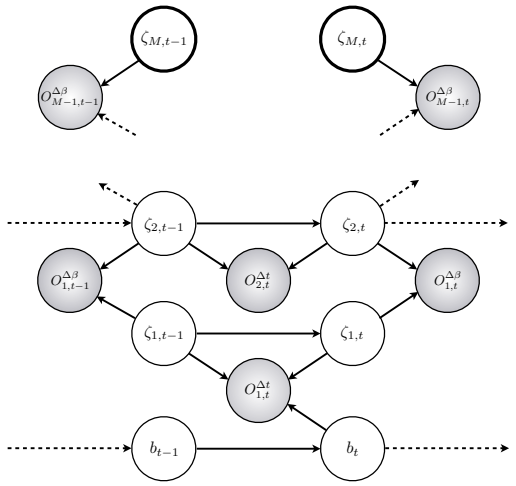
$q_{i,t}$	RBM at inverse temp. β_i at time-step t. $q_{i,t}(x) = \tilde{q}_{i,t}(x)/Z_{i,t}$, with $i \in [1, M]$.
θ	set of model parameters.
$F_{i,t}(x)$	free-energy assigned by $q_{i,t}$ to input configuration x.
$\zeta_{i,t}$	log partition function of model $q_{i,t}$.
$\mathcal{X}_{i,t}$	mini-batch of samples $\{x_{i,t}^{(n)} \sim q_{i,t}(x); n \in [1, N]\}$.
$\mathcal{Y}_t$	additional mini-batch of samples $\{x_{1,t}^{(n)} \sim q_{1,t}(x); n \in [1, N_Y]\}, N_Y \geq N$.
$\mathcal{D}$	set of training examples.
$\mu_{t,t}, P_{t,t}$	estimated mean and covariance of posterior distribution $p(\zeta_t \mid O_{t:0}^{(\Delta t)}, O_{t:0}^{(\Delta \beta)})$.
Σ_ζ	fixed covariance matrix of $p(\zeta_t \mid \zeta_{t-1})$.
$\epsilon_{init}, \epsilon_t$	initial learning rate and rate at time t.
$s_{i,t}$	coarse estimate of $Z_{i+1,t}/Z_{i,t}$ used to generate bridging distribution q^*.
$q_{i,t}^*$	bridging distribution with $q_{i,t}^*(x) = \frac{\tilde{q}_{i,t}(x)\tilde{q}_{i+1,t}(x)}{s_{i,t}\tilde{q}_{i,t}(x)+\tilde{q}_{i+1,t}(x)}$.
$r(q_1, q_2, x_1, x_2)$	swap probability used by PT, with $r_{i,t} = \min\left(1, \frac{\tilde{q}_1(\mathbf{x}_2)\tilde{q}_2(\mathbf{x}_1)}{\tilde{q}_1(\mathbf{x}_1)\tilde{q}_2(\mathbf{x}_2)}\right)$.

1 Parallel Tempering Algorithm

We divide our algorithm into three parts. Algorithm 1 presents the pseudo-code for the Parallel Tempering sampling algorithm. For details, we refer the reader to [1].

Algorithm 1 `sample_PT` $(q_{:,t}, \mathcal{X}_{:,t-1}, k)$

Initialize $\mathcal{X}_{i,t}$ as empty sets, $\forall i \in [1, M]$.
for $n \in [1, N]$ **do**
  for $i \in [1, M]$ **do**
    Initialize Markov chain associated with model $q_{i,t}$ with state $x_{i,t-1}^{(n)}$.
    Perform k steps of Gibbs sampling, yielding $x_{i,t}^{(n)}$.
  end for
  Swap $x_{i,t}^{(n)} \leftrightarrow x_{i+1,t}^{(n)}$ with prob. $r(q_{i,t}, q_{i+1,t}, x_{i,t}^{(n)}, x_{i+1,t}^{(n)})$, $\forall$ even i.
  Swap $x_{i,t}^{(n)} \leftrightarrow x_{i+1,t}^{(n)}$ with prob. $r(q_{i,t}, q_{i+1,t}, x_{i,t}^{(n)}, x_{i+1,t}^{(n)})$, $\forall$ odd i.
  $\mathcal{X}_{i,t} \leftarrow \mathcal{X}_{i,t} \cup \{x_{i,t}^{(n)}\}$, $\forall i$.
end for
return $\mathcal{X}_{:,t}$.



System Equations:

$$p(\zeta_0) = \mathcal{N}(\mu_0, \Sigma_0)$$

$$p(\zeta_t \mid \zeta_{t-1}) = \mathcal{N}(\zeta_{t-1}, \Sigma_\zeta)$$

$$p(O_t^{(\Delta t)} \mid \zeta_t, \zeta_{t-1}) = \mathcal{N}(C[\zeta_t, \zeta_{t-1}]^T\ ,\ \Sigma_{\Delta t})$$

$$p(O_t^{(\Delta\beta)} \mid \zeta_t) = \mathcal{N}(H\zeta_t, \Sigma_{\Delta\beta})$$

$$C = \begin{bmatrix} & 1 & & 0 \\ & 0 & & 0 \\ I_M & \vdots & -I_M & \vdots \\ & 0 & & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} -1 & +1 & 0 & 0 & & 0 \\ 0 & -1 & +1 & 0 & \vdots & 0 \\ & & \cdots & & & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 \end{bmatrix}$$

Figure 1: A directed graphical model for log partition function tracking. The shaded nodes represent observed variables, and the double-walled nodes represent the tractable $\zeta_{M,:}$ with $\beta_M = 0$. For clarity of presentation, we show the bias term as distinct from the other $\zeta_{i,t}$ (recall $b_t = \zeta_{M+1,t}$.)

Inference Equations:

(i) $p\left(\zeta_{t-1}, \zeta_t \mid O^{(\Delta t)}_{t-1:0}, O^{(\Delta\beta)}_{t-1:0}\right) = \mathcal{N}(\eta_{t-1,t-1}, V_{t-1,t-1})$
with $\eta_{t-1,t-1} = \begin{bmatrix} \mu_{t-1,t-1} \\ \mu_{t-1,t-1} \end{bmatrix}$ and $V_{t-1,t-1} = \begin{bmatrix} P_{t-1,t-1} & P_{t-1,t-1} \\ P_{t-1,t-1} & \Sigma_\zeta + P_{t-1,t-1} \end{bmatrix}$

(ii) $p(\zeta_{t-1}, \zeta_t \mid O^{(\Delta t)}_{t:0}, O^{(\Delta\beta)}_{t-1:0}) = \mathcal{N}(\eta_{t,t-1}\ ,\ V_{t,t-1})$
with $V_{t,t-1} = (V^{-1}_{t-1,t-1} + C^T\Sigma^{-1}_{\Delta t}C)^{-1}$ and $\eta_{t,t-1} = V_{t,t-1}(C^T\Sigma_{\Delta t}O^{(\Delta t)}_t + V^{-1}_{t-1,t-1}\eta_{t-1,t-1})$

(iii) $p\left(\zeta_t \mid O^{(\Delta t)}_{t:0}, O^{(\Delta\beta)}_{t-1:0}\right) = \mathcal{N}\left(\mu_{t,t-1}\ ,\ P_{t,t-1}\right)$ with $\mu_{t,t-1} = [\eta_{t,t-1}]_2$ and $P_{t,t-1} = [V_{t,t-1}]_{2,2}$

(iv) $p(\zeta_t \mid O^{(\Delta t)}_{t:0}, O^{(\Delta\beta)}_{t:0}) = \mathcal{N}(\mu_{t,t}, P_{t,t})$
with $P_{t,t} = (P^{-1}_{t,t-1} + H^T\Sigma^{-1}_{\Delta\beta}H)^{-1}$ and $\mu_{t,t} = P_{t,t}(H^T\Sigma_{\Delta\beta}O^{(\Delta\beta)}_t + P^{-1}_{t,t-1}\mu_{t,t-1})$

Figure 2: Inference equations for our log partition tracking algorithm, a variant on the Kalman filter. For any vector v and matrix V, we use the notation $[v]_2$ to denote the vector obtained by preserving the bottom half elements of v and $[V]_{2,2}$ to indicate the lower right-hand quadrant of V.

2 Kalman Filter

Algorithm 2 presents the tracking algorithm. The statistical estimate $O^{(\Delta t)}_{i,t}$ is computed as an average of importance weights, measured between adjacent models $q_{i,t}$ and $q_{i,t-1}$. The estimate $O^{(\Delta\beta)}_{i,t}$ is computed through bridge sampling, applied to models $q_{i+1,t}$ and $q_{i,t}$. These observations are combined through a Kalman filter, which also exploits a smoothness prior on the evolution of ζ_t.

We include the graphical model, system equations and inference equations in Figures 1 & 2 for completeness. We however refer the reader to the accompanying paper for a more thorough description of these figures.

3 Simultaneous Tracking and Learning

Finally, Algorithm 3 ties everything together, performing joint training and estimation of the log partition function. Note that using two sets of samples from the target distribution ($\mathcal{X}_{1,t}$ and $\mathcal{Y}_t$)

Algorithm 2 `kalman_filter` $(q_{:,t-1}, q_{:,t}, \mu_{t-1,t-1}, P_{t-1,t-1}, s_{i,t}, \Sigma_\zeta)$

Using $\mu_{t-1,t-1}$, $P_{t-1,t-1}$ and Σ_ζ, compute $\eta_{t-1,t-1}$ and $V_{t-1,t-1}$ through equation (i).

for $i \in [1, M]$ **do**

$$w_{i,t}^{(n)} \leftarrow \frac{\tilde{q}_{i,t}\left(x_{i,t-1}^{(n)}\right)}{\tilde{q}_{i,t-1}\left(x_{i,t-1}^{(n)}\right)}; \quad O_{i,t}^{(\Delta t)} \leftarrow \log\left[\frac{1}{N}\sum_{n=1}^{N} w_{i,t}^{(n)}\right]; \quad \Sigma_{\Delta t} \leftarrow \mathrm{Diag}\left[\frac{\mathrm{Var}[w_{i,t}]}{\left(\sum_n w_{i,t}^{(n)}\right)^2}\right]$$

end for

Using $O_{i,t}^{(\Delta t)}$ and $\Sigma_{\Delta t}$, compute $\eta_{t,t-1}$ and $V_{t,t-1}$. through equation (ii).
Compute $\mu_{t,t-1}$ and $P_{t,t-1}$ using equation (iii).

for $i \in [1, M-1]$ **do**

$$u_{i,t}^{(n)} \leftarrow \frac{q_{i,t}^*\left(x_{i,t}^{(n)}\right)}{\tilde{q}_{i,t}\left(x_{i,t}^{(n)}\right)}; \quad v_{i,t}^{(n)} \leftarrow \frac{q_{i,t}^*\left(x_{i+1,t}^{(n)}\right)}{\tilde{q}_{i+1,t}\left(x_{i+1,t}^{(n)}\right)}$$

$$O_{i,t}^{(\Delta\beta)} \leftarrow \log\frac{1}{N}\sum_{n=1}^{N} u_{i,t}^{(n)} - \log\frac{1}{N}\sum_{n=1}^{N} v_{i,t}^{(n)}$$

$$\Sigma_{\Delta\beta} \leftarrow \mathrm{Diag}\left[\frac{\mathrm{Var}[u_{i,t}]}{\left(\sum_n u_{i,t}^{(n)}\right)^2} + \frac{\mathrm{Var}[v_{i,t}]}{\left(\sum_n v_{i,t}^{(n)}\right)^2}\right]$$

end for

Using $O_{i,t}^{(\Delta\beta)}$ and $\Sigma_{\Delta\beta}$, compute $\eta_{t,t}$ and $V_{t,t}$ through equation (iv).
Return $(\eta_{t,t}, V_{t,t})$.

is not required. Their use is inspired from [2] and allows us to separately tune N, the number of "tempered" mini-batches and N_Y, the size of the mini-batch used to estimate the gradient.

Algorithm 3 `main`

Initialize θ_1 and compute exact log partition functions $\zeta_{:,1}$.
Initialize $\mu_{1,1}$ with $\zeta_{:,1}$, $P_{1,1}[1:M, 1:M] = 0$ and $P_{1,1}[M+1, M+1] = \epsilon_{init} \cdot \sigma_b^2$.
Initialize samples $\mathcal{X}_{:,1}$ and $\mathcal{Y}_{:,1}$ according to the RBM visible biases.
Initialize $s_{i,1}$ to $\exp(\zeta_{i+1,1} - \zeta_{i,1})$, $\forall i \in [1, M-1]$.

for $t \in [2, T]$ **do**

Obtain training examples $\mathcal{X}_t^+ = \{x^{(n)} \in \mathcal{D}; n \in [1, N]\}$

$$\theta_t \leftarrow \theta_{t-1} - \epsilon_t \left(\frac{1}{N}\sum_{x \in \mathcal{X}_t^+}\left[\frac{\partial F(x;\theta_t)}{\partial\theta}\right] - \frac{1}{N}\sum_{y \in \mathcal{Y}_t}\left[\frac{\partial F(y;\theta_{F,t})}{\partial\theta}\right]\right).$$

Choose N samples from $\mathcal{Y}_t$ to swap with $\mathcal{X}_{1,t}$.

$\mathcal{X}_{:,t} \leftarrow$ `sample_PT` $(q_{:,t}, \mathcal{X}_{:,t-1}, k)$.

$\mathcal{Y}_t \leftarrow$ `sample_Gibbs` $(q_{1,t}, \mathcal{Y}_{t-1}, k)$.

$(\mu_{t,t}, P_{t,t}) \leftarrow$ `kalman_filter` $(q_{:,t-1}, q_{:,t}, \mu_{t-1,t-1}, P_{t-1,t-1}, s_{i,t}, \Sigma_\zeta)$

$\hat{\zeta}_{:,t} \leftarrow \mu_{t,t}; \quad s_{i,t+1} \leftarrow \exp(\hat{\zeta}_{i+1,t} - \hat{\zeta}_{i,t}), \forall i \in [1, M-1].$

end for

References

[1] Desjardins, G., Courville, A., Bengio, Y., Vincent, P., and Delalleau, O. (2010). Tempered Markov chain monte carlo for training of restricted Boltzmann machine. In *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 145–152.

[2] Salakhutdinov, R. (2010). Learning deep boltzmann machines using adaptive mcmc. In L. Bottou and M. Littman, editors, *Proceedings of the Twenty-seventh International Conference on Machine Learning (ICML-10)*, volume 1, pages 943–950. ACM.