Twisted Variational Sequential Monte Carlo

Dieterich Lawson
New York University
jdl404@nyu.edu

George Tucker
Google Brain
gjt@google.com

Christian A. Naesseth
Linköping University
christian.a.naesseth@liu.se

Chris J. Maddison
University of Oxford, DeepMind
cmaddis@stats.ox.ac.uk

Ryan P. Adams
Princeton University
rpa@princeton.edu

Yee Whye Teh
University of Oxford, DeepMind
y.w.teh@stats.ox.ac.uk

Abstract

Latent variable models have achieved wide-ranging success in generative modeling of images, speech, and video. Recent work has developed specialized objectives for fitting sequential latent variable models [1, 2, 3], but these methods are based on filtering sequential Monte Carlo (SMC), which can prevent them from successfully modeling data with long-term dependencies. In this work, we introduce twisted variational sequential Monte Carlo (TVSMC), a family of variational objectives that use learned tilting functions to approximate smoothing SMC without direct access to the smoothing distributions. We demonstrate that models trained with TVSMC can outperform models trained with VSMC, and theoretically, the TVSMC bound can become tight.

1 Introduction

Latent variable models are an expressive class of generative model which have been successfully used to model natural images [4, 5], video [6, 7], and speech [8, 9]. In this work, we focus on learning parametric sequential latent variable models of the form

$$p(x_{1:T}, z_{1:T}) = p(z_1)p(x_1|z_1)\prod_{t=2}^{T} p(z_t|x_{1:t-1}, z_{1:t-1})p(x_t|x_{1:t-1}, z_{1:t}),$$

where $x_{1:T}$ is a sequence of observations and $z_{1:T}$ is a sequence of continuous latent variables.

Ideally, we would fit these models by maximizing the log marginal likelihood $\log p(x_{1:T})$, but this requires an intractable marginalization over the latent variables. Instead, we maximize a lower bound constructed from an unbiased estimator of the marginal likelihood, $\hat{Z}(x_{1:T})$,

$$\log p(x_{1:T}) = \log \mathbb{E}\left[\hat{Z}(x_{1:T})\right] \geq \mathbb{E}\left[\log \hat{Z}(x_{1:T})\right], \tag{1}$$

which follows from Jensen's inequality. The ELBO is the classic choice for this lower bound where $\hat{Z}(x_{1:T})$ is defined by a single importance weight [10, 11, 12]. The authors of [1] showed that the variance of the estimator is closely related to the tightness of its corresponding bound, so developing bounds based on lower-variance estimators can yield better objectives. Recent work has developed variational objectives based on multiple importance sampling [13] and filtering SMC [1, 2, 3].

SMC is a general procedure that takes as input unnormalized "target" distributions $\{\gamma(z_{1:t})\}_{t=1}^{T}$ and a set of proposal distributions $\{q(z_t|z_{1:t})\}_{t=1}^{T}$. As long as $\gamma(z_{1:T}) = p_\theta(x_{1:T}, z_{1:T})$, then SMC returns an unbiased marginal likelihood estimator (under mild conditions). SMC's key benefit is its resampling operation, where particle locations are sampled from an approximation to the normalized target distributions. When the normalized targets are close to the posterior, resampling can reduce

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

variance. If the distributions are mismatched, however, then trajectories likely under the posterior but unlikely under the target distributions may be discarded, potentially increasing variance.

In filtering SMC, the unnormalized targets are the unnormalized filtering distributions, $p(z_{1:t}, x_{1:t})$, so resampling reduces variance when the filtering distributions are close to the posterior $p(z_{1:t}|x_{1:T})$. As a result, if $z_{1:t}$ is not conditionally independent of $x_{t+1:T}$ given $x_{1:t}$, then a bound based on filtering SMC can never be tight [1]. Unfortunately, in many popular models this independence structure does not hold, such as bidirectional or hierarchical latent variable models (e.g. [14, 15]).

Choosing $\gamma(z_{1:t}) = p(x_{1:T}, z_{1:t})$ would resolve this issue, but computing $p(x_{1:T}, z_{1:t})$ is intractable. To overcome this, we use learned tilting functions to approximate these *smoothing* distributions. We call our approach *twisted variational sequential Monte Carlo* (TVSMC). This allows our bound to become tight (theoretically) at the cost of solving a more difficult optimization problem. We propose several techniques to optimize this objective, and show that TVSMC outperforms VSMC on a toy problem. Finally, we conduct preliminary experiments with complex nonlinear models, which reveal remaining challenges in optimizing the tilting functions.

2 Learning the Smoothing Distributions

The goal of TVSMC is to optimize the variational objective (eq. (1)) constructed from the marginal likelihood estimator of a smoothing SMC algorithm. The key issue is that the unnormalized smoothing distributions $p(x_{1:T}, z_{1:t})$ are required for smoothing SMC, but are generally intractable. We do have access to $p(x_{1:t}, z_{1:t})$, however, so our approach is to learn a parametric tilting function $r(x_{t+1:T}|x_{1:t}, z_{1:t})$ such that $p(x_{1:t}, z_{1:t})r(x_{t+1:T}|x_{1:t}, z_{1:t}) \approx p(x_{1:T}, z_{1:t})$. Running SMC with these γ will approximate smoothing SMC when $r(x_{t+1:T}|x_{1:t}, z_{1:t})$ is close to the "lookahead" distributions, $p(x_{t+1:T}|x_{1:t}, z_{1:t})$.

Let $\widehat{Z}_{1:T}(x_{1:T})$ be the normalizing constant estimate returned from running SMC with unnormalized targets $\gamma(z_{1:t}) = p(x_{1:t}, z_{1:t})r(x_{t+1:T}|x_{1:t}, z_{1:t})$, proposal distributions $q(z_t|z_{1:t-1}, x_{1:T})$, and K particles. We define the TVSMC objective as

$$\mathcal{L}_K^{\text{TVSMC}}(p, q, r, x_{1:T}) = \mathbb{E}\left[\log \widehat{Z}_T(x_{1:T})\right]. \tag{2}$$

$\mathcal{L}_K^{\text{TVSMC}}(p, q, r, x_{1:T})$ is a lower bound on $\log p(x_{1:T})$, and when $r(x_{t+1:T}|x_{1:t}, z_{1:t}) = p(x_{t+1:T}|x_{1:t}, z_{1:t})$ and $q(z_t|z_{1:t-1}, x_{1:T}) = p(z_t|z_{1:t-1}, x_{1:T})$, the bound is tight.

2.1 Optimizing the Unified Objective

One way to optimize eq. (2) is stochastic gradient ascent jointly in the parameters of p, q, and r. This algorithm is conceptually similar to VSMC — the only difference is the change to the target distributions caused by introducing r.

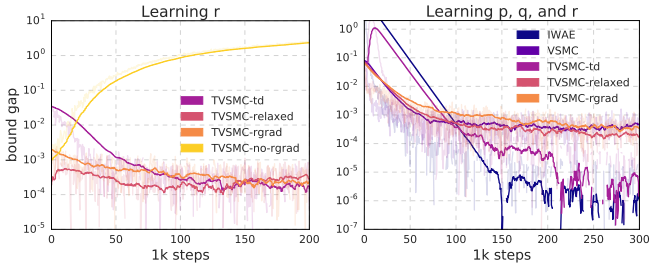
The VSMC papers omit gradients from the discrete resampling operations due to high variance, but our experiments below show that incorporating resampling gradient information is crucial for learning the parameters of r. One option is to use the score function estimator of the gradient of eq. (2) and to attempt to reduce its variance with control variates. We call this method TVSMC-rgrad, for details see the Appendix. Unfortunately, this approach scales poorly to large problems, so it mostly serves as a baseline.

An alternative is to use the Gumbel-Softmax relaxed gradient estimator [16, 17]. We call this method TVSMC-relaxed, for more details see the Appendix. This approach can potentially scale to large problems with complex models.

2.2 Temporal Difference Learning

Guarniero et al. [18] and Heng et al. [19] introduced an alternative approach for learning r: use temporal difference (TD) learning to fit an approximation of a distribution related to the lookahead distributions, and then obtain r from that approximation when needed. The TD learning approach is motivated by a recursive decomposition of $p(x_{t:T}|x_{1:t-1}, z_{1:t})$ in terms of the same distribution shifted one step forward in time,

$$p(x_{t:T}|x_{1:t-1}, z_{1:t}) = p(x_t|x_{1:t-1}, z_{1:t})\mathbb{E}_{z_{t+1}\sim p(z_{t+1}|x_{1:t}, z_{1:t})}\left[p(x_{t+1:T}|x_{1:t}, z_{1:t+1})\right]. \tag{3}$$



(a) (b)

Dataset	TVSMC-td	VSMC
Nottingham	-2.85	**-2.68**
JSB	**-6.90**	**-6.90**
MuseData	-6.64	**-6.20**
Piano-midi	-7.82	**-7.76**
TIMIT	51,194	**59,058**

(c)

Figure 1: Left, Middle: Training curves for the Gaussian diffusion task. The vertical axis is the bound gap in nats, i.e. the true likelihood of the data under the model minus the lower bound value. Right: Test-set lower bounds for VRNNs fit to various datasets. The bounds for the TIMIT dataset are in nats per sequence, all other bounds are nats per timestep. For each method the bound that was used for training is reported. VSMC numbers are from [1].

All terms except distributions of the form $p(x_{t:T}|x_{1:t-1}, z_{1:t})$ are directly available from the model definition. Let $\{\tilde{r}_t(x_{1:T}, z_{1:t})\}_{t=1}^{T-1}$ be a sequence of learnable positive functions which we will use to approximate the unknown distributions $\{p(x_{t:T}|x_{1:t-1}, z_{1:t})\}_{t=1}^{T-1}$. Substituting $\tilde{r}_t(x_{1:T}, z_{1:t})$ for $p(x_{t:T}|x_{1:t-1}, z_{1:t})$ and taking the log of both sides of (3) gives a series of Bellman equations, allowing us to fit $\tilde{r}_t$ by minimizing the squared Bellman error. We solve this optimization problem by stochastic gradient descent (SGD), interleaving SGD steps for p and q with updates to $\tilde{r}$. When needed, r can be obtained by integrating $\tilde{r}$ against the transition density of the model. We call this method TVSMC-td, for details see the Appendix.

3 Experiments

To evaluate the different approaches for learning the parameters of r, we constructed a linear 1-D Gaussian diffusion task where we expected learning r would be essential. The model $p(x, z_{1:T})$ factors as $\mathcal{N}(z_1; 0, 1)\left(\prod_{t=2}^{T}\mathcal{N}(z_t; z_{t-1} + b_{t-1}, 1)\right)\mathcal{N}(x; z_T + b_T, 1)$ where x is observed, $z_{1:T}$ are latent, and $b_{1:T}$ are parameters in the model. We expected VSMC to perform poorly because the filtering distributions $p(z_t|z_{t-1})$ contain no information about x, suggesting that resampling should only degrade the samples. For these, experiments our variational family and lookahead distributions were parameterized by learned linear functions, which includes the true posterior and lookahead distribution. For experimental details see the Appendix.

Figures 1a and 1b show the results of using the proposed methods to perform inference and learning in this model. Figure 1a suggests that it is possible to learn r effectively, but that without resampling gradients, training quickly diverges. Figure 1b shows it is possible to jointly learn p, q, and r, and that of the TVSMC methods, TVSMC-td converges the fastest.

We also evaluated TVSMC's ability to train complex nonlinear models by using it to fit variational recurrent neural networks (VRNNs) [8] on four piano sheet-music datasets [20] and the TIMIT speech dataset [21] (Figure 1c; see Appendix for details). We evaluated only TVSMC-td in this setting because it was the most promising method in the simple experiments. Although in principle TVSMC should improve upon VSMC, in this case, TVSMC underperforms VSMC. Because VSMC can be understood as a specialization of TVSMC where $r = 1$, this result indicates issues with optimizing r.

4 Conclusion

We introduced Twisted Variational Sequential Monte Carlo, a flexible family of variational objectives for learning in sequential latent variable models. We explored several ways to optimize this objective, and provided empirical evidence in favor of TVSMC-td. In future work we will investigate better optimization procedures for r and apply TVSMC to new models.

References

[1] Chris J Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6573–6583, 2017.

[2] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 968–977, 2018.

[3] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. In *International Conference on Learning Representations*, 2018.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *ICLR*, 2014.

[5] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 2014.

[6] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

[7] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[8] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *NIPS*, 2015.

[9] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *NIPS*, 2016.

[10] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2017.

[11] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[12] Andriy Mnih and Danilo J Rezende. Variational inference for Monte Carlo objectives. *arXiv preprint arXiv:1602.06725*, 2016.

[13] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *ICLR*, 2016.

[14] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746, 2016.

[15] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models. *arXiv preprint arXiv:1702.08396*, 2017.

[16] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[18] Pieralberto Guarniero, Adam M Johansen, and Anthony Lee. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647, 2017.

[19] Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential monte carlo. *arXiv preprint arXiv:1708.08396*, 2017.

[20] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *ICML*, 2012.

[21] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. DARPA TIMIT acoustic phonetic continuous speech corpus CDROM, 1993.

[22] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[23] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning.

5 Appendix

5.1 Experimental Details

5.1.1 1-D Gaussian Diffusion

For these experiments we defined our variational family as $q_t(z_t|z_{t-1}, x) = \mathcal{N}(z_t; f_t(z_{t-1}, x), \sigma_t^2)$ where $\{f_t\}_{t=1}^T$ are learnable linear functions and $\{\sigma_t^2\}_{t=1}^T$ are learnable constants. Similarly, r is parameterized as $r(x|z_t) = \mathcal{N}(x; g_t(z_t), \tau_t^2)$ where $\{g_t\}_{t=1}^T$ are learnable linear functions and $\{\tau_t^2\}_{t=1}^T$ are learnable constants. Note that this variational family includes the true posterior over z_t, and the parametric family for r includes the true marginal distribution $p(x|z_t)$. The data were sampled from the model, and T was set to 10. The $b_{1:T}$ in the data generating process were set by sampling them from the interval $[0, 1]$. Optimization was performed with ADAM [22] with learning rate 0.0001.

5.1.2 Pianorolls and TIMIT

The models in these experiments had one observation for each latent variable, unlike the simple Gaussian diffusion task which had only one observation. Because of this, a naive implementation of the lookahead distribution $r(x_{t+1:T}|z_{1:t}, x_{1:t})$ would score all future observations at every timestep, leading to an $O(T^2)$ complexity for evaluating the bound.

To avoid this in the TVSMC-td method, we allow $\log \tilde{r}$ to be a quadratic function of z_t, where the coefficients are produced by parametric functions (deep neural networks) of $z_{1:t-1}$ and $x_{1:T}$. When the coefficient-producing functions are recurrent neural networks (RNNs) that take as input $z_{1:t-1}$ and $x_{1:T}$, the state at a given timestep can be reused to compute the coefficients at the next timestep. This allows for evaluating the bound in $O(T)$ time.

For the pianoroll experiments we trained single-layer variational recurrent neural networks (VRNNs) with factorial Gaussian latent states and Bernoulli emission distributions. For the Nottingham, MuseData, and Piano-midi.de datasets we used a latent state of size 64, and for JSB we used a latent state of size 32. The variational posterior was defined by a neural networks of two layers of the same size as the latent state. $\tilde{r}$ was defined by a two-layer neural network that accepted as input a bidirectional RNN state run over the inputs $x_{1:T}$ and the current state of the VRNN (which is a function of $z_{1:t}$). The bidirectional RNN was one layer of the same size as the latent state.

Optimization was performed with ADAM, and a learning rate was selected based on validation performance from a grid search over $\{1 \times 10^{-5}, 3 \times 10^{-5}, 1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}\}$.

For TIMIT the experimental setup was similar, except the emission distributions were Gaussian, a latent state of size 256 was used, and the variational posterior and lookahead distribution neural networks were correspondingly larger. Our experimental setup was based on the code released by the authors of [1], available at github.com/tensorflow/models/tree/master/research/fivo. The code uses the TensorFlow framework [23].

5.2 Gradient

Let p, q, and r be parameterized differentiably by θ, let q be from a reparameterizable family, and let all latents z_t^k in algorithm 1 be reparameterized. Also assume a fixed set of timesteps on which resampling events occur, $R \subseteq \{1, \ldots, T\}$. Then eq. (2) has the gradient

$$\mathbb{E}\left[\nabla_\theta \log \widehat{Z}_T(x_{1:T}) + \sum_{t \in R} \left(\log \widehat{Z}_T(x_{1:T}) - \log \widehat{Z}_t(x_{1:T})\right) \sum_{k=1}^K \nabla_\theta \log \overline{w}_t^{a_t^k}\right]. \tag{4}$$

Algorithm 1 Sequential Monte Carlo

procedure SMC($\{\gamma(z_{1:t}), q(z_t|z_{1:t-1})\}_{t=1}^T, K$)
  $w_0^{1:K} = 1, \quad \widehat{Z}_0 = 1$
  for $t = 1, \ldots, T$ **do**
    for $k = 1, \ldots, K$ **do**
      $z_t^k \sim q_t(z_t|z_{1:t-1}^k)$
      $z_{1:t}^k = (z_{1:t-1}^k, z_t^k)$
      $\alpha_t^k = \gamma(z_{1:t}^k)/(\gamma(z_{1:t-1}^k)q(z_t^k|z_{1:t-1}^k))$
      $w_t^k = w_{t-1}^k \alpha_t^k$
    end for
    $\widehat{Z_t/Z_{t-1}} = \left(\sum_{k=1}^K w_t^k\right) / \sum_{k=1}^K w_{t-1}^k$
    $\widehat{Z}_t = \widehat{Z}_{t-1}(\widehat{Z_t/Z_{t-1}})$
    if should resample **then**
      $z_{1:t}^{1:K} = \text{RESAMPLE}(z_{1:t}^{1:K}, w_t^{1:K})$
      $w_t^{1:K} = 1$
    end if
  end for
  return $\widehat{Z}_{1:T}$, $z_{1:T}^{1:K}$
end procedure

Algorithm 2 Multinomial Resampling

procedure RESAMPLE($z_{1:t}^{1:K}, w_t^{1:K}$)
  for $k = 1, \ldots, K$ **do**
    $a_t^k \sim \text{Categorial}(\overline{w}_t^{1:K})$
    $\tilde{z}_{1:t}^k = z_{1:t}^{a_t^k}$
  end for
  return $\tilde{z}_{1:t}^{1:K}$
end procedure

Algorithm 3 Relaxed Resampling

procedure RESAMPLE($z_{1:t}^{1:K}, w_t^{1:K}, \tau$)
  for $k = 1, \ldots, K$ **do**
    $a_t^k \sim \text{Concrete}(\overline{w}_t^{1:K}, \tau)$
    $\tilde{z}_{1:t}^k = \sum_{j=1}^K (a_t^k)_j z_{1:t}^j$
  end for
  return $\tilde{z}_{1:t}^{1:K}$
end procedure

as derived in [1].

5.3 Relaxed Resampling

When estimating eq. (4) with relaxed resampling, the Categorical distribution in the resampling step of SMC is replaced with a Concrete distribution on the $(K-1)$-dimensional simplex. Samples from the Concrete distribution are K-vectors instead of discrete ancestor indices, so to define "soft" inheritance we set the inherited state to be the convex combination of the parent states defined by the sampled vector. At evaluation we sample discrete indices. When using relaxed resampling gradients the score-function gradients in equation (4) are not computed.

5.4 TD Learning

To re-iterate what was stated in the main text, the TD learning approach is motivated by a recursive decomposition of $p(x_{t:T}|x_{1:t-1}, z_{1:t})$ in terms of the same distribution shifted one step forward in time,

$$p(x_{t:T}|x_{1:t-1}, z_{1:t}) = p(x_t|x_{1:t-1}, z_{1:t})\mathbb{E}_{z_{t+1}\sim p(z_{t+1}|x_{1:t}, z_{1:t})}\left[p(x_{t+1:T}|x_{1:t}, z_{1:t+1})\right]. \quad (5)$$

All terms except distributions of the form $p(x_{t:T}|x_{1:t-1}, z_{1:t})$ are directly available from the model definition. Let $\{\tilde{r}_t(x_{1:T}, z_{1:t})\}_{t=1}^{T-1}$ be a sequence of learnable positive functions which we will use to approximate the unknown distributions $\{p(x_{t:T}|x_{1:t-1}, z_{1:t})\}_{t=1}^{T-1}$. Substituting $\tilde{r}_t(x_{1:T}, z_{1:t})$ for $p(x_{t:T}|x_{1:t-1}, z_{1:t})$ and taking the log of both sides of (5) gives Bellman equations for $t = 1, \ldots T-1$,

$$\log \tilde{r}_t(x_{1:T}, z_{1:t}) = \log p(x_t|x_{1:t-1}, z_{1:t}) + \log \mathbb{E}_{z_{t+1}\sim p(z_{t+1}|x_{1:t}, z_{1:t})}\left[\tilde{r}_{t+1}(x_{1:T}, z_{1:t+1})\right] \quad (6)$$

with $\tilde{r}_T(x_{1:T}, z_{1:T}) = p(x_T|x_{1:T-1}, z_{1:T})$ as it is available from the model. The authors of [19] and [18] propose fitting $\tilde{r}_t$ by minimizing the squared Bellman error of (6), i.e. the left hand side minus the right hand side squared.

Computing the Bellman error of (6) requires computing the expectation of $\tilde{r}_t$ with respect to the transition density of the model. We can compute this integral analytically when $\log \tilde{r}(x_{1:T}, z_{1:t})$ and $\log p(z_t|x_{1:t-1}, z_{1:t-1})$ are quadratic in z_t.

We obtain r_t from $\tilde{r}_t$ by computing the same expectation. This is motivated by the decomposition

$$p(x_{t:T}|x_{1:t-1}, z_{1:t-1}) = \mathbb{E}_{z_t\sim p(z_t|x_{1:t-1}, z_{1:t-1})}\left[p(x_{t:T}|x_{1:t-1}, z_{1:t})\right] \quad (7)$$

which states that when $\tilde{r}_t$ satisfies the Bellman equations and equals $p(x_{t:T}|x_{1:t-1}, z_{1:t})$, the lookahead distributions $p(x_{t:T}|x_{1:t-1}, z_{1:t-1})$ can be obtained by integrating $\tilde{r}_t$ against the transition density of the model. For further details see [19, 18].