

Bayesian Grammar Learning for Inverse Procedural Modeling

- Supplementary Material -

Anđelo Martinović and Luc Van Gool

1. Calculation of Acceptance Probabilities in rjMCMC

The state of the Markov chain is defined as a grammar derivation $\delta = (\tau, \alpha)$, where α is the parameter vector of variable dimension. In every iteration, the chain is evolved by applying a diffusion move, which does not change the dimensionality of the parameter vector, or a jump move, which adds or removes elements from the parameter vector.

1.1. Diffusion Move

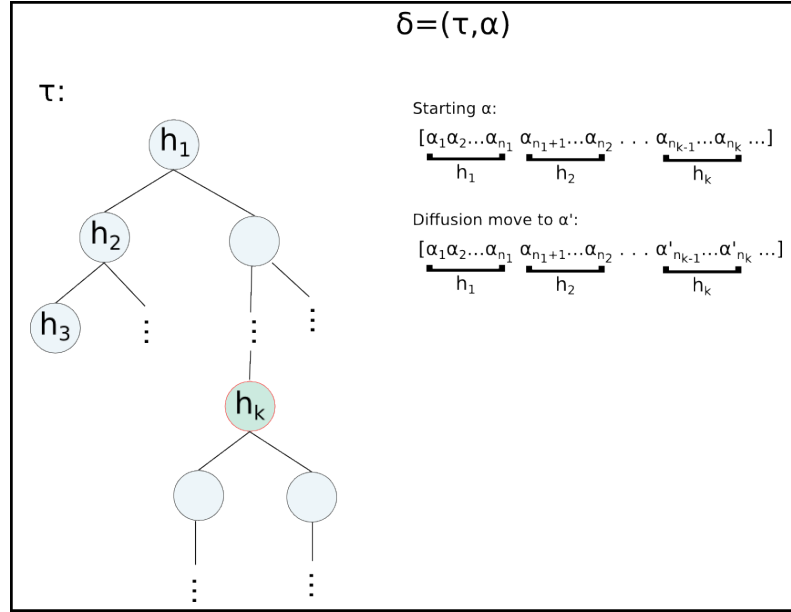


Figure 1. Diffusion move.

A random node h_k is selected in the tree. The parameters corresponding to that node in the tree are then resampled from independent Gaussian proposal distributions. The parameters at other nodes remain unchanged. The acceptance probability for a move from state x to state y is given by [1]:

$$\rho_{x \rightarrow y} = \min\{1, \frac{p(y)q(x|y)}{p(x)q(y|x)}\}$$

In our case, the acceptance probability for a diffusion move from a derivation δ to δ' is:

$$\rho_{\delta \rightarrow \delta'} = \min\{1, \frac{p(\delta'|I)q(\delta|\delta')}{p(\delta|I)q(\delta'|\delta)}\}$$

Since the derivation tree remains unchanged, we can write the proposal distribution q as:

$$\begin{aligned}
q(\delta'|\delta) &= q(\alpha'|\alpha) \\
&= q(\alpha'_{n_{k-1}}|\alpha_{n_{k-1}})q(\alpha'_{n_{k-1}+1}|\alpha_{n_{k-1}+1}) \dots q(\alpha'_{n_k}|\alpha_{n_k}) \\
&= \mathcal{N}(\alpha'_{n_{k-1}}; \alpha_{n_{k-1}}, \sigma^2) \cdot \mathcal{N}(\alpha'_{n_{k-1}+1}; \alpha_{n_{k-1}+1}, \sigma^2) \cdot \dots \cdot \mathcal{N}(\alpha'_{n_k}; \alpha_{n_k}, \sigma^2) \\
&= \mathcal{N}(\alpha_{n_{k-1}}; \alpha'_{n_{k-1}}, \sigma^2) \cdot \mathcal{N}(\alpha_{n_{k-1}+1}; \alpha'_{n_{k-1}+1}, \sigma^2) \cdot \dots \cdot \mathcal{N}(\alpha_{n_k}; \alpha'_{n_k}, \sigma^2) \\
&= q(\alpha_{n_{k-1}}|\alpha'_{n_{k-1}})q(\alpha_{n_{k-1}+1}|\alpha'_{n_{k-1}+1}) \dots q(\alpha_{n_k}|\alpha'_{n_k}) \\
&= q(\alpha|\alpha') \\
&= q(\delta|\delta')
\end{aligned}$$

The proposal distribution is symmetric, so the acceptance rate simplifies to:

$$\rho_{\delta \rightarrow \delta'} = \min\{1, \frac{P(\delta'|I)}{P(\delta|I)}\} = \min\{1, e^{-(E_{\delta'} - E_{\delta})}\}$$

1.2. Jump Move

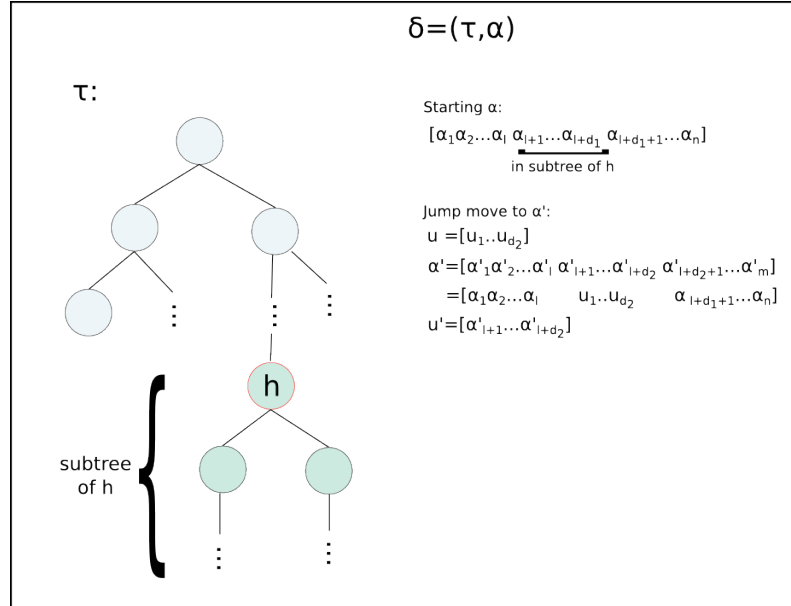


Figure 2. Jump move.

In the jump move, a random node h is selected from the tree, and a new rule is sampled from all rules applicable to the current LHS. If the size of the RHS changes, the entire subtree of h has to be rederived. This changes the topology of the tree, as well as the dimension of the parameter vector.

Following the discussion in [2], we denote the current state of the chain with $x = (k, \alpha)$, where k denotes the current space of dimension n_k and α the current set of parameters. From the current state, a move m is attempted by sampling a random vector u of r_m random numbers from a known density g_m . The new state $x' = (k', \alpha')$ can be constructed with a deterministic function h such that $(x', u') = h_m(x, u)$. Here, u' is a r' -dimensional vector of random numbers sampled from a known density g' , needed for the reverse move from x' to x : $(x, u) = h'(x', u')$. The transformation between (x, u) to (x', u') must be a diffeomorphism, i.e. both h and its inverse h' are required to be bijections and differentiable. This leads to a 'dimension-matching' constraint: $n_k + r_m = n_{k'} + r'_m$.

The acceptance rate for the move m is given by:

$$\rho_{x \rightarrow x'} = \min\left\{1, \frac{\pi(x')}{\pi(x)} \frac{j_m(x')}{j_m(x)} \frac{g'_m(u')}{g_m(u)} \left| \frac{\partial(\alpha', u')}{\partial(\alpha, u)} \right| \right\} \quad (1)$$

where π represents the target density, $j_m(x)$ is the probability that the move m is attempted at state x , and the last factor represents the Jacobian of the mapping from (α, u) to (α', u') .

We shall now define our jump move. Since in our case the state of the chain is defined as $\delta = (\tau, \alpha)$, the dimension of the initial space n_k is the dimension of the parameter vector α . Let $l + 1$ be the index of the first parameter of node h in the concatenated vector α , d_1 the number of parameters in the subtree τ_h underneath the node h , and d_2 the number of parameters in the subtree τ'_h after resampling the rule at h . We can define u and u' as vectors of d_2 and d_1 uniformly sampled numbers in the interval $[0, 1]$, respectively. In terms of the previous discussion, $r_m = d_2$, $r'_m = d_1$, and $g_m(u) = \mathcal{U}_{[0,1]}(u)$.

We can now fully write the mapping h_m as follows (Eq. 10 in the main paper [3])

$$\begin{aligned} \alpha'_i &= \begin{cases} \alpha_i & , i \in [1, l] \\ u_{i-l} & , i \in [l+1, l+d_2] \\ \alpha_{i-d_2+d_1} & , i \in [l+d_2+1, m] \end{cases} \\ u'_i &= \alpha_{i+l} \quad , i \in [1, d_2] \end{aligned} \quad (2)$$

The reverse mapping is obtained from Eq. 2 by swapping (α, u, m, d_2) with (α', u', n, d_1) . It is clear that by using this mapping, the Jacobian in Eq. 1 reduces to unity. Since the proposal distributions g_m are uniform on the interval $[0, 1]$, the third term in Eq. 1 also vanishes. We define the probability of the move j_m from $\delta = (\tau, \alpha)$ to $\delta' = (\tau', \alpha')$ as:

$$j(\delta'|\delta) = q_\tau(h) \prod_{s \in \tau'_h} P(r_s)$$

where $q_\tau(h)$ is the probability of selecting a nonterminal h in the tree τ . The second term evaluates the probabilities of rules selected in the derivation of the newly derived subtree τ'_h .

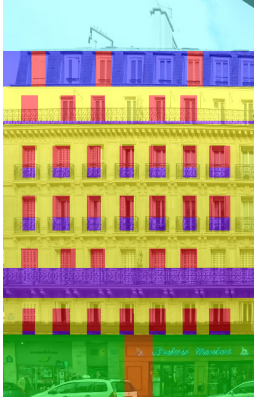
Now we can write the final acceptance probability as:

$$\begin{aligned} \rho_{\delta \rightarrow \delta'} &= \min\left\{1, \frac{P(\delta'|I)}{P(\delta|I)} \frac{j(\delta|\delta')}{j(\delta'\delta)}\right\} \\ &= \min\left\{1, \frac{P(\delta'|I)}{P(\delta|I)} \frac{q_{\tau'}(h) \prod_{s \in \tau_h} P(r_s)}{q_\tau(h) \prod_{s \in \tau'_h} P(r_s)}\right\} \\ &= \min\left\{1, \frac{P(I|\delta') \prod_{s \in \tau'} P(r_s) \prod_{s \in \tau'} \phi(A(r_s))}{P(I|\delta) \prod_{s \in \tau} P(r_s) \prod_{s \in \tau} \phi(A(r_s))} \frac{q_{\tau'}(h) \prod_{s \in \tau_h} P(r_s)}{q_\tau(h) \prod_{s \in \tau'_h} P(r_s)}\right\} \\ &= \min\left\{1, \frac{q_{\tau'}(h)}{q_\tau(h)} \frac{P(I|\delta') \prod_{s \in \tau'} \phi(A(r_s))}{P(I|\delta) \prod_{s \in \tau} \phi(A(r_s))}\right\} \\ &= \min\left\{1, \frac{q_{\tau'}(h)}{q_\tau(h)} \frac{e^{-E_{\delta'}^{image}} \cdot e^{-E_{\delta'}^{attr}}}{e^{-E_{\delta'}^{image}} \cdot e^{-E_{\delta'}^{attr}}}\right\} \\ &= \min\left\{1, \frac{q_{\tau'}(h)}{q_\tau(h)} e^{-[(E_{\delta'}^{img} + E_{\delta'}^{attr}) - (E_{\delta}^{img} + E_{\delta}^{attr})]}\right\} \end{aligned}$$

where we have utilized Eq. 6 and Eq. 7 from the main paper [3] to factorize the posterior and to write the final expression in terms of energies instead of probabilities.

2. Additional Results

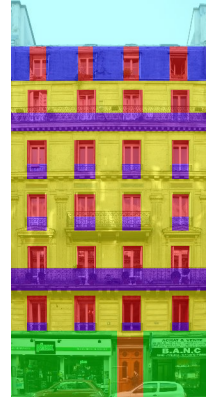
We show some additional results obtained by parsing images from one cross-validation fold on the ECP dataset.



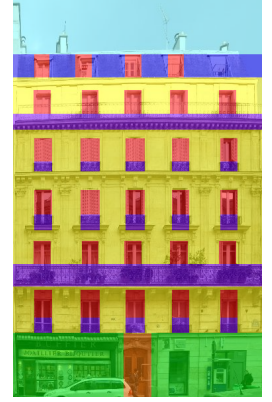
monge_1



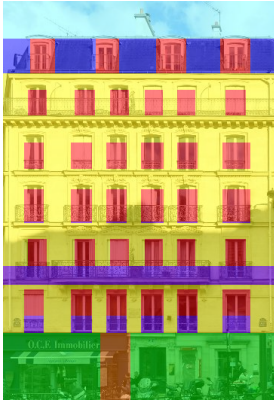
monge_3



monge_6



monge_9



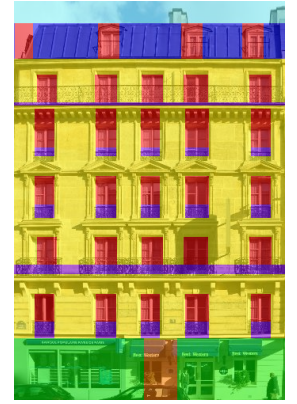
monge_29bis



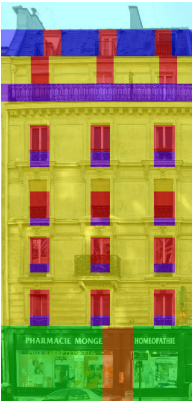
monge_61



monge_65



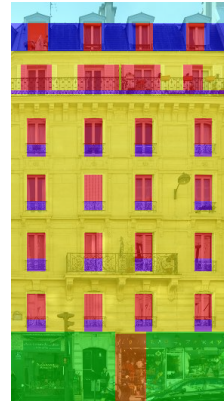
monge_71



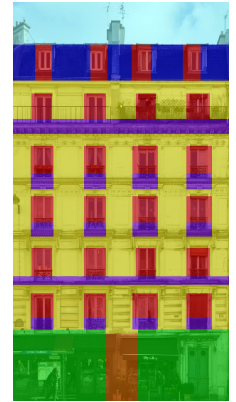
monge_74



monge_79bis



monge_86



monge_101

Acknowledgement

This work was supported by ERC Advanced Grant VarCity and Research Programme of the Fund for Scientific Research-Flanders (Belgium) (FWO - G.0004.08).

References

- [1] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, 1995. 1
- [2] P. J. Green and D. I. Hastie. Reversible jump mcmc. http://www.stats.bris.ac.uk/~mapjg/papers/rjmc_20090613.pdf, 2009. 2
- [3] A. Martinović and L. Van Gool. Bayesian Grammar Learning for Inverse Procedural Modeling. In *CVPR*, 2013. To appear. 3