Supplementary Material for Iterative Channel Estimation for Discrete Denoising under Channel Uncertainty

Hongjoon Ahn[1] **and Taesup Moon**[1,2]
[1] Department of Artificial Intelligence, [2]Department of Electrical and Computer Engineering,
Sungkyunkwan University, Suwon, Korea 16419
{hong0805,tsmoon}@skku.edu

1 Proof of Lemma 1

Consider the maximization problem of (Eq.(14), Manuscript) using $Q(x^n)$ as in (Eq.(18), Manuscript) to obtain the updated $\mathbf{\Pi}^{(t+1)}$ from $\mathbf{\Pi}^{(t)}$.

$$\begin{aligned}\mathbf{\Pi}^{(t+1)} =&\arg\max_{\mathbf{\Pi}} \sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)}) \log \frac{p(x^n,Z^n;\mathbf{\Pi})}{Q(x^n;\mathbf{\Pi}^{(t)})}\\ =&\arg\max_{\mathbf{\Pi}} \sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)}) \log p(x^n,Z^n;\mathbf{\Pi})\\ =&\arg\max_{\mathbf{\Pi}} \sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)}) \log p_{\mathcal{X}}(x^n)p(Z^n|x^n;\mathbf{\Pi})\\ =&\arg\max_{\mathbf{\Pi}} \sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)}) \log \prod_{i=1}^{n}\mathbf{\Pi}(x_i,Z_i)\\ =&\arg\max_{\mathbf{\Pi}} \sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)})\\ &\cdot\sum_{i=1}^{n}\sum_{j=1}^{|\mathcal{X}|}\sum_{\ell=1}^{|\mathcal{Z}|}\mathbb{1}_{\{x_i=j,Z_i=\ell\}}\log\mathbf{\Pi}(j,\ell) \qquad (1)\end{aligned}$$

Since above maximization process has following constraint, $\sum_{\ell=1}^{|\mathcal{Z}|}\mathbf{\Pi}_{j\ell}=1$, we can consider the Lagrangian of (1)

$$\begin{aligned}L(\mathbf{\Pi},\boldsymbol{\lambda}) =&\sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)})\\ &\cdot\sum_{i=1}^{n}\sum_{j=1}^{|\mathcal{X}|}\sum_{\ell=1}^{|\mathcal{Z}|}\mathbb{1}_{\{x_i=j,Z_i=\ell\}}\log\mathbf{\Pi}(j,\ell)\\ &+\sum_{j=1}^{|\mathcal{X}|}\boldsymbol{\lambda}_j(1-\sum_{\ell=1}^{|\mathcal{Z}|}\mathbf{\Pi}(j,\ell)) \qquad (2)\end{aligned}$$

Then, to apply the KKT condition, the partial derivatives of the Lagrangian w.r.t $\mathbf{\Pi}^{(t)}(j,\ell)$ and $\boldsymbol{\lambda}_j$ becomes

$$\begin{aligned}\frac{\partial L}{\partial\mathbf{\Pi}(j,\ell)} =&\frac{1}{\mathbf{\Pi}(j,\ell)}\sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)})\\ &\cdot\sum_{i=1}^{n}\mathbb{1}_{\{x_i=j,Z_i=\ell\}}-\boldsymbol{\lambda}_j\\ \frac{\partial L}{\partial\boldsymbol{\lambda}_j} =&1-\sum_{\ell=1}^{|\mathcal{Z}|}\mathbf{\Pi}_{j\ell}. \qquad (3)\end{aligned}$$

Now, assume that the parameter satisfying KKT condition is $\mathbf{\Pi}^{(t+1)}$. Then,

$$\mathbf{\Pi}^{(t+1)}(j,\ell) = \frac{1}{\boldsymbol{\lambda}_j}\sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)})\sum_{i=1}^{n}\mathbb{1}_{\{x_i=j,Z_i=\ell\}} \qquad (4)$$

Using (3) and (4),

$$\begin{aligned}\boldsymbol{\lambda}_j =&\sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)})\sum_{i=1}^{n}\mathbb{1}_{\{x_i=j\}}\\ \therefore\mathbf{\Pi}^{(t+1)}(j,\ell) =&\frac{\sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)})\sum_{i=1}^{n}\mathbb{1}_{\{x_i=j,Z_i=\ell\}}}{\sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)})\sum_{i=1}^{n}\mathbb{1}_{\{x_i=j\}}} \qquad (5)\end{aligned}$$

However, we can change (5) to an expression that contains (Eq.(16), Manuscript).

$$
\begin{aligned}
&\sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)}) \sum_{i=1}^{n} \mathbb{1}_{\{x_i=j,Z_i=\ell\}} \\
&= \sum_{i=1}^{n} \sum_{x^n} \mathbb{1}_{\{x_i=j,Z_i=\ell\}} Q(x^n;\mathbf{\Pi}^{(t)}) \\
&= \sum_{i=1}^{n} \sum_{x^n} \mathbb{1}_{\{x_i=j,Z_i=\ell\}} \prod_{\ell=1}^{n} Q(x_l|Z^n;\mathbf{\Pi}^{(t)}) \\
&= \sum_{i=1}^{n} \sum_{x_i} \mathbb{1}_{\{x_i=j,Z_i=\ell\}} Q(x_i|Z^n;\mathbf{\Pi}^{(t)}) \\
&= \sum_{i=1}^{n} \mathbb{1}_{\{Z_i=\ell\}} Q(x_i=j|Z^n;\mathbf{\Pi}^{(t)})
\end{aligned}
$$

Likewise,

$$
\sum_{x^n} Q(x^n;\mathbf{\Pi}^{(t)}) \sum_{i=1}^{n} \mathbb{1}_{\{x_i=j\}} = \sum_{i=1}^{n} Q(x_i=j|Z^n;\mathbf{\Pi}^{(t)})
$$

$$
\therefore \mathbf{\Pi}^{(t+1)}(j,\ell) = \frac{\sum_{i=1}^{n} \mathbb{1}_{\{Z_i=\ell\}} Q(x_i=j|Z^n;\mathbf{\Pi}^{(t)})}{\sum_{i=1}^{n} Q(x_i=j|Z^n;\mathbf{\Pi}^{(t)})}
$$

2 Experimental Details

2.1 Output dimension reduction

As a separate contribution, we also address one additional limitation of N-DUDE. Namely, the original N-DUDE has an output size of $|\mathcal{S}| = |\hat{\mathcal{X}}|^{|\mathcal{Z}|}$, which can quickly grow very large when the alphabet size of the data grows. For example, even for DNA sequence that has alphabet size of 4, the output size of $\mathbf{p}^k(\mathbf{w},\cdot)$ becomes $|\mathcal{S}| = 4^4 = 256$ as shown in Figure 1. Such exponential growth of the output size may cause overfitting and inaccurate approximation for the induced posterior (Eq.(9), Manuscript).

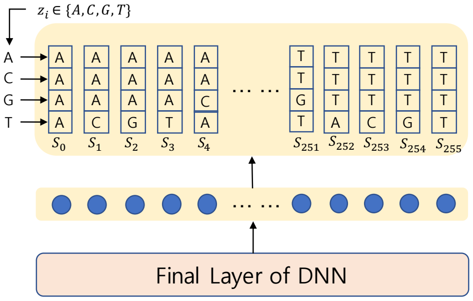
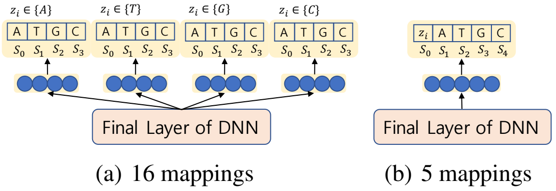


Figure 1: The original output layer of N-DUDE for DNA data.

In order to make ICE-N-DUDE more scalable with large alphabet size, we considered two output dimension reduction methods as shown in Figure 2, shown with the DNA example. First, Figure 2(a) shows reducing the output size to $|\hat{\mathcal{X}}||\mathcal{Z}| = 16$ by implementing $|\mathcal{Z}|$ different output layers having $|\mathcal{X}|$ outputs. Note all 256 mappings in Figure 1 can be enumerated by combining the *partial* mappings for each Z_i given in Figure 2(a). Second, Figure 2(b) shows further reducing the output size to $|\hat{\mathcal{X}}| + 1 = 5$. That is, by simplifying the denoising to either "saying-what-you-see" (i.e., $s(Z_i) = Z_i$) or "saying-one-in-$\hat{\mathcal{X}}$, we can work with this reduced output size. With this reduction, the unnecessary variance in the model could reduce and the summation in (Eq.(9), Manuscript) would always involve only two mappings, hence, the approximation quality of $D_{KL}(\tilde{Q}(x^n;\mathbf{\Pi}^{(t)})\|p(x^n|Z^n;\mathbf{\Pi}^{(t)}))$ could improve. In fact, in our experimental results, we observed the second reduction yields much better denoising as well as the channel estimation results, hence, all of our results regarding N-DUDE employ the second reduction structure.



(a) 16 mappings (b) 5 mappings

Figure 2: The reduced output layer of N-DUDE for DNA data.

2.2 Training details for binary image denoising

In the binary image denoising experiment, we used the first 10 images in PASCAL and all 8 images for Standard for estimating the channel before carrying out the denoising in each set. For ICE-N-DUDE and ICE-CUDE, the estimated channel by ICE was plugged-in, and the network parameters were fine-tuned for each image *separately*. For a fair comparison, N-DUDE($\mathbf{\Pi}$) and CUDE($\mathbf{\Pi}$) were also first trained with the same images as ICE and BW, before fine-tuning for each image.

2.3 True channels

We used three different asymmetric $\mathbf{\Pi}$'s in binary image denoising experiments with each $\mathbf{\Pi}$ having an average noise level of 0.1, 0.2, and 0.3.

$$
\mathbf{\Pi}_{0.1} = \begin{bmatrix} 0.88 & 0.12 \\ 0.09 & 0.91 \end{bmatrix}
$$

$$
\mathbf{\Pi}_{0.2} = \begin{bmatrix} 0.83 & 0.17 \\ 0.23 & 0.77 \end{bmatrix}
$$

$$
\mathbf{\Pi}_{0.3} = \begin{bmatrix} 0.72 & 0.28 \\ 0.33 & 0.67 \end{bmatrix}
$$

For the DNA experiment, we used 4×4 asymmetric $\mathbf{\Pi}$ as below.

$$\mathbf{\Pi}_{DNA} = \begin{bmatrix} 0.8122 & 0.0034 & 0.0894 & 0.0950 \\ 0.0096 & 0.8237 & 0.0808 & 0.0859 \\ 0.1066 & 0.0436 & 0.7774 & 0.0724 \\ 0.0704 & 0.0690 & 0.0889 & 0.7717 \end{bmatrix}$$