Nonparametric Bayesian Dictionary Learning for Machine Listening

Dawen Liang
Electrical Engineering
dl2771@columbia.edu

1 Introduction

Machine listening, i.e., giving machines the ability to extract useful information from the acoustic world in a manner similar to listeners, is a relatively undeveloped field that presents many interesting challenges. In the real world, sound rarely comes from a single source. For example, a piece of music may contain voices from the singers and accompaniments from different instruments, or a recording of a meeting may sequentially consist of speech from different speakers. In this project, I aim to treat sound signals – one-dimensional time series – as mixtures of acoustically-meaningful events. A time-dependent dictionary learning approach based on nonparametric Bayesian is proposed.

Dictionary learning in general has been increasingly popular in speech and audio signal processing recently as it provides a compact representation for complex and structured sound signals. The basic idea behind dictionary learning is to learn a dictionary and project the signal onto this dictionary space: $\boldsymbol{x}_i = \mathbf{D} \cdot \boldsymbol{w}_i + \varepsilon_i$. Here $\boldsymbol{x}_i$ is a vector of length F, representing information from a short frame of the original signal, with $i \in \{1, 2, \cdots, N\}$ being frame indices. In speech and audio signal processing, the typical signal representation is spectrogram, a time (frame)-frequency matrix. An example spectrogram for a clip of speech is visualized in Figure 1. The x-axis represents the time evolution of the signal in the unit of short frame. The frequency components for each frame is the column of the spectrogram, which is $\boldsymbol{x}_i$ following the notation introduced above. Thus spectrogram can be mathematically denoted as $\mathbf{X} = [\boldsymbol{x}_1|\boldsymbol{x}_2|\cdots|\boldsymbol{x}_N] \in \Re^{F\times N}$. The color indicates the relative energy within the corresponding frequency bin. Matrix $\mathbf{D} = [\boldsymbol{d}_1|\boldsymbol{d}_2|\cdots|\boldsymbol{d}_K] \in \Re^{F\times K}$ is the dictionary with K items, and $\boldsymbol{w}_i \in \Re^K$ are the coordinates in the K-dimensional dictionary space. ε_i is modeled as Gaussian noise. Normally, K is larger than F, the dimension of $\boldsymbol{x}_i$, thus a sparse representation can be obtained, i.e. most of the entries in $\boldsymbol{w}_i$ are 0. Research [4] has shown that the dictionary learned from speech data corresponds to phonemes. In the music domain, [2] also shows that such sparse representations help improve the performance on genre classification tasks.

An important problem with current work is that the temporal information is largely ignored. Researchers in auditory perception research[1] have argued that time in audition plays a similar role to space in vision in determining the structure internal to auditory objects. However, most existing work treats short frames $\boldsymbol{x}_i$ (with durations typically around 50ms) independently without considering temporal information. The model proposed in this project aims to incorporate temporal information within the general nonparametric Bayesian dictionary learning.

2 Related Work and Proposed Improvement

2.1 Nonparametric Bayesian Dictionary Learning

The basic idea behind nonparametric Bayesian modeling is that the complexity of the model grows with the amount of observed data, i.e., the complexity of the model is potentially infinite. In our case, this means that we will set K to a large value and let the inference algorithm decide the

[1] http://plato.stanford.edu/entries/perception-auditory/

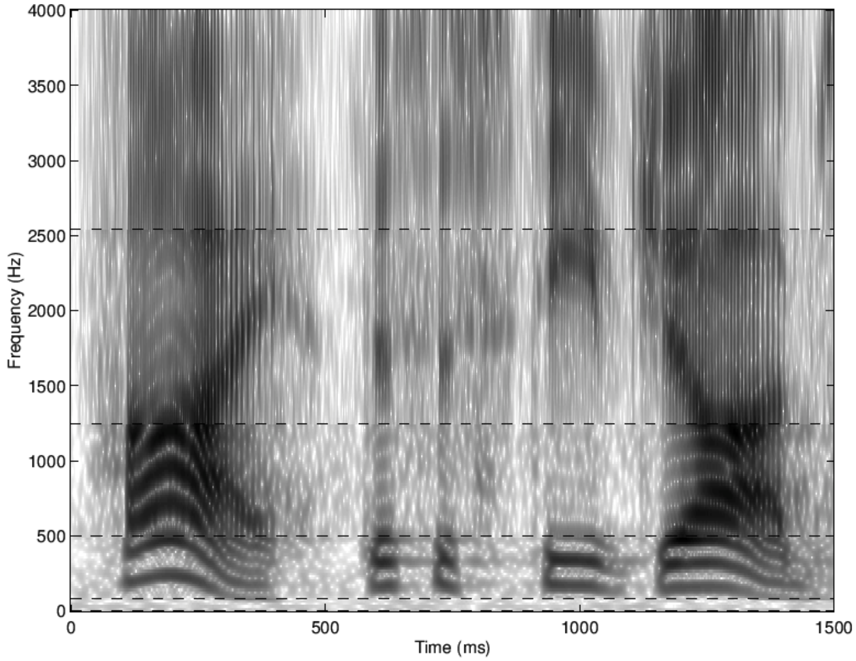

Figure 1: Spectrogram of a short clip of speech (from JOS).

proper number of dictionary items to use. There is some existing work on nonparametric Bayesian dictionary learning with application to image processing [7], from which I got the inspiration for this project. The nonparametric Bayesian dictionary learning uses the similar formulation as $\boldsymbol{x}_i = \mathbf{D} \cdot \boldsymbol{w}_i + \varepsilon_i$. However, instead of directly calculating coordinate matrix $\mathbf{W} = [\boldsymbol{w}_1|\boldsymbol{w}_2|\cdots|\boldsymbol{w}_N] \in \Re^{K\times N}$, it is decomposed as the Hadamard product of 2 matrices $\mathbf{S} \odot \mathbf{Z}$, where $\mathbf{S} \in \Re^{K\times N}$ is a real-valued matrix which acts as the weights for dictionary items, and $\mathbf{Z} \in \{0,1\}^{K\times N}$ is a random binary matrix which is constructed from Beta/Bernoulli Processes [5]. The $\mathbf{Z}$ drawn from Beta/Bernoulli Process imposes sparsity to the original weights, as it masks some of the weights by turning the corresponding elements to 0. More on this in Section 3.1.

Inference algorithms based on Gibbs sampling or variational Bayes have been proposed in earlier work [7, 5, 6]. It has been shown [7] that for the task of image denoising and inpainting, the Gibbs sampler can run just as efficiently as some parametric alternatives, like K-SVD [1], while achieving better performance.

2.2 Temporal Constraints

Since all the existing work on nonparametric Bayesian dictionary learning applies to stationary data, like images, x_i mostly represents a image patch and $i \in \{1, 2, \cdots, N\}$ has *no* sequential information embedded. This is reflected in the model that if viewing it as a generative process, every $\boldsymbol{s}_i$ is drawn independently from the remaining $\boldsymbol{s}_{\neg i}$. To add temporal constraints, it is natural to add a 1st-order Markov assumption, i.e., instead of drawing $\boldsymbol{s}_i$ from a certain distribution independently, we draw $\boldsymbol{s}_i$ conditioned on the value of $\boldsymbol{s}_{i-1}$. This can also be observed from the spectrogram in Figure 1 that there are a lot of continuities going on horizontally, which indicates a strong time dependency.

It's worth noticing that in neuroscience, there has been some ongoing research on visual and auditory perception, among which the hierarchical spike coding of sound [3] introduced the idea of modeling sound based on both coarse and fine scales. The 1st-order Markov assumption is in fact a fine scale modeling, which suggests that as part of the future improvement, we can also incorporate the coarse scale information of sound.

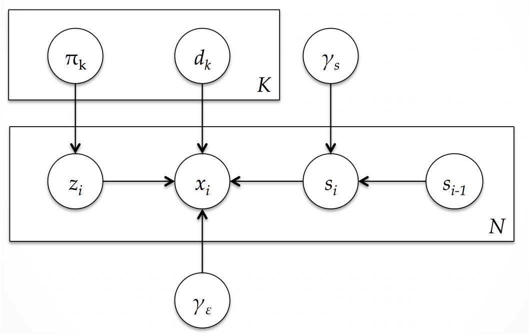

Figure 2: The graphical model plate notation for BPTDL.

3 Nonparametric Bayesian Dictionary Learning

3.1 Beta Process

I will not put any formal definition of Beta Process here as the definition in [5] is quite well-formed. An intuitive way to interpret Beta Process is that it generates a vector of infinite length with each of the element i.i.d. drawn from a Beta distribution. Since Beta distribution is the conjugate prior of Bernoulli distribution (Binomial distribution with only one trial), a corresponding Bernoulli Process can be defined, from which a binary vector can be randomly drawn.

A finite approximation of Beta/Bernoulli Process is constructed exactly the same way following this intuition:

$$\begin{aligned} z_{ik} &\sim \text{Bernoulli}(\pi_k) \\ \pi_k &\sim \text{Beta}(\frac{a}{K}, \frac{b(K-1)}{K}) \end{aligned}$$

Here z_{ik} is the element at ith row and kth column of the $\mathbf{Z}$ matrix defined in Section 2. The $\boldsymbol{\pi}$ vector enforces sparseness to the $\mathbf{Z}$ matrix, and this will act as the building block for the proposed model.

3.2 Beta Process Time-dependent Dictionary Learning (BPTDL)

The proposed model (Beta Process Time-dependent Dictionary Learning, abbreviated as BPTDL) can be formally defined as follows with the graphical model plate notation shown in Figure 2:

$$\begin{aligned} \boldsymbol{x}_i &= \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i) + \epsilon_i \\ \boldsymbol{d}_k &\sim \mathcal{N}(0, F^{-1}\mathbf{I}_F) \\ z_{ik} &\sim \text{Bernoulli}(\pi_k) \\ \pi_k &\sim \text{Beta}(\frac{a_0}{K}, \frac{b_0(K-1)}{K}) \\ \boldsymbol{s}_i &\sim \mathcal{N}(\boldsymbol{s}_{i-1}, \gamma_s^{-1}\mathbf{I}_K) \\ \epsilon &\sim \mathcal{N}(0, \gamma_\epsilon^{-1}\mathbf{I}_F) \\ \gamma_s &\sim \Gamma(c_0, d_0) \\ \gamma_\epsilon &\sim \Gamma(e_0, f_0) \end{aligned}$$

The noninformative priors are placed on hyperparameters $\{\gamma_s, \gamma_\epsilon\}$. Experiments shows that the choice of a_0 and b_0 won't lead to too much difference as long as K is large enough.

As for the inference, Gibbs sampler is used as it's straightforward to write down the posterior distribution. All the updating rules for each variable are derived in Appendix B.

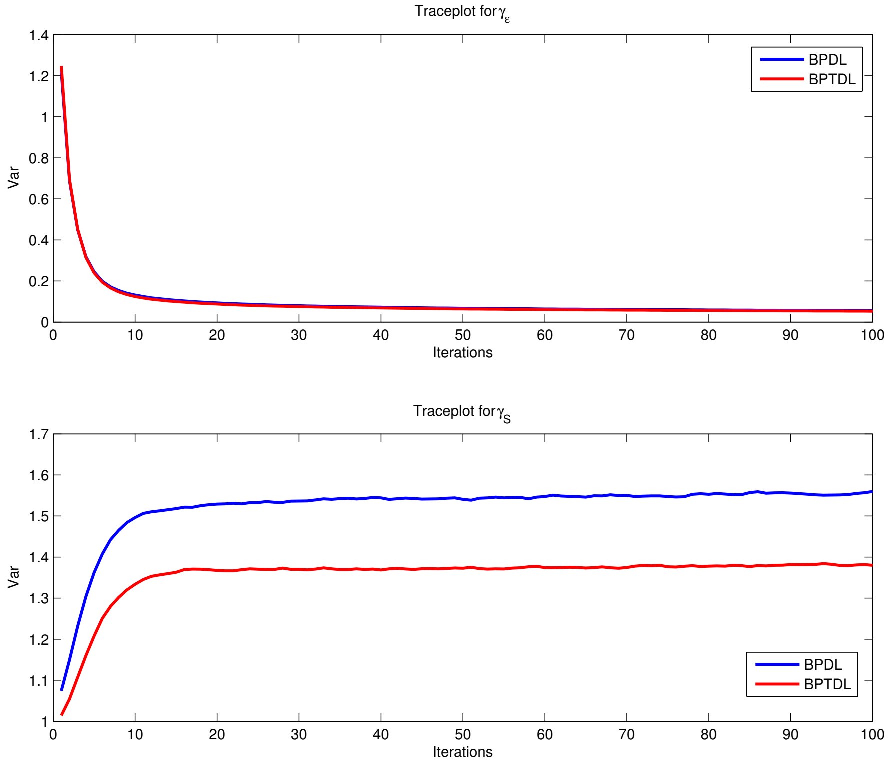

Figure 3: The traceplots for γ_ϵ (upper) and γ_s (lower) from both BPDL and BPTDL.

4 Experiment

Three sets of experiments have been conducted on sound of different kinds: The first one is the comparison between BPTDL and Beta Process Dictionary Learning with no time constrain (short as BPDL, which is exactly the same method in [7] applied to spectrogram) on a recording of woodwind quintet. The second one is a naïve denoising test on the same woodwind quintet recording. The last one is dictionary learning on a meeting recording. All of the experiments showed promising results.

4.1 BPTDL v.s. BPDL

Dictionary learning is performed on a woodwind quintet (the performance of a group of 5 woodwind instruments) with both BPTDL and BTDL for $K = 256$ and a FFT size of 512 with 50% frame overlap. Note that in this case, K actually equals to F. Thus the representation learnd is not very sparse – around 80 out of 256 dictionary items are frequently used for BPDL and around 70 out of 256 for BPTDL. Some future experiments will include the ones with larger K, say 1024. But for now we are only comparing the performance between BPDL and BPTDL, and there is still something interesting showing up.

Figure 3 shows the traceplots for γ_ϵ and γ_s (The plots are actually the standard derivation $\sigma_\epsilon = \sqrt{1/\gamma_\epsilon}$ and $\sigma_s = \sqrt{1/\gamma_s}$ for better visualization, especially for γ_ϵ which can be large when reaching convergence). As we can see, both methods inferred a very small σ_ϵ around 0.05 with the one from BPTDL slightly smaller. This makes sense as the recording we are testing on is basically clean. The inferred σ_s differs, with the one from BPTDL smaller (1.38 to 1.56), which indicates that BPTDL does try to capture the variance for the current frame given the previous one, instead of a steady 0 (as in BPDL) which is more likely to lead to a larger variance in order to fit the data. On the other hand, this result also proves the justifiability of the time-dependent model assumption.

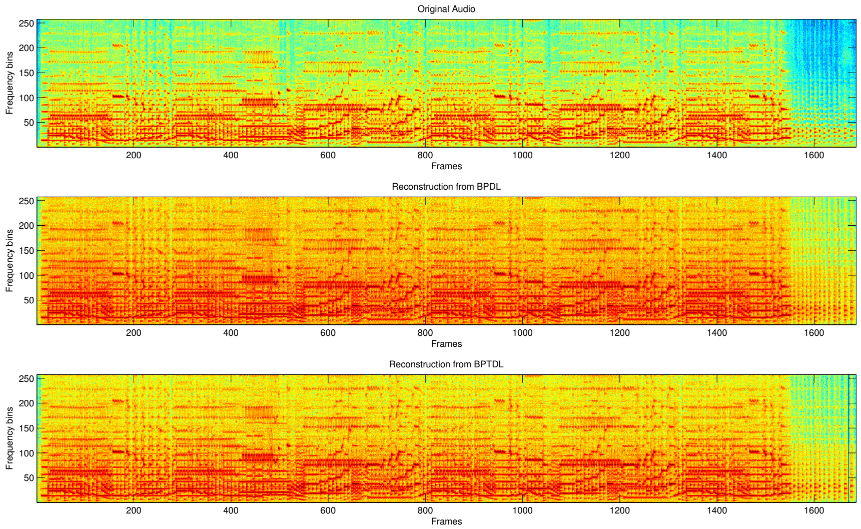

Figure 4: The spectrogram for the original audio (upper), reconstruction from BPDL (middle), and reconstruction from BPTDL (lower).

Another sanity check I did is to reconstruct the original audio from $\mathbf{D}(\mathbf{S} \odot \mathbf{Z})$. The result spectrograms are shown in Figure 4. Both reconstructions sound very close to the original one, as the inferred σ_ϵ is very small. However, the difference in spectrogram which is plotted in Decibel is more obvious – the reconstruction from BPTDL has slightly better performance on high frequency than the one from BPDL. Note that as the reconstructed $\mathbf{D}(\mathbf{S} \odot \mathbf{Z})$ may contain non-positive values (we didn't put any constrain on how the dictionary or weights should behave), while the true spectrogram should always to positive as it represents the energy distribution, here I "cheated" when plotting the spectrogram by ignoring all the non-positive values. This also gives me clue about some future work that positive constrain should be part of the model specification.

4.2 Denoising

In speech and audio signal processing, the noisy signal for denoising research is usually constructed by adding a clean signal with noises of some form in time domain and then denoising is done in frequency domain. The denoising I performed, on the other hand, is very naïve as I just added random noise in the frequency domain. These 2 approaches are in general not equivalent as we missed out the phase information when directly adding noise on the frequency domain. However, the reason I did this is that if I want to have the noise captured by the ϵ term in the model, then it has to be normal distribution with 0 mean and σ_ϵ standard deviation, which is too simplified for the real-world noise. The denoising with my simplified assumption works with no surprise (even when the noise with standard deviation of 5, compared to a standard deviation of 1.38 for clean recording), as it perfectly follows my model assumption. If real-world denoising task is our goal, then we need to investigate the more sophisticated signal+noise distribution and incorporate it into the existing model.

4.3 Meeting recording

One of the proposed application for the model is to better facilitate sound source separation, as we hope acoustically-meaningful dictionary can be learned. In this experiment, a clip of a meeting recording from *ICSI Meeting Recorder project* [2] is used with 6 different speakers and a lot of overlap between different speakers. The speeches are recorded at 16 kHZ sampling rate, and a frame length of 32 ms (FFT size of 512 and 50% overlap) is chosen as it's suggested that a frame length close to

[2] `http://www1.icsi.berkeley.edu/Speech/mr/mtgrcdr.html`

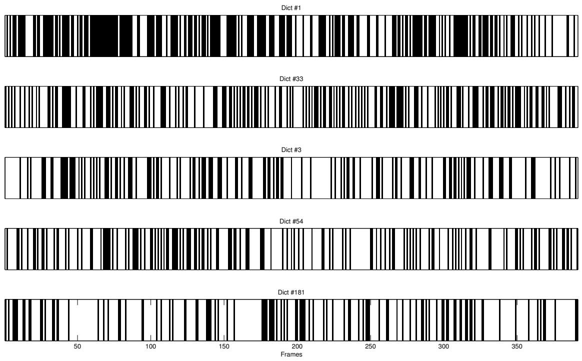

Figure 5: The activation from matrix $\mathbf{Z}$ for the top 5 most frequently used dictionary items.

20 ms works best for human speech. For the dictionary, K is set to 256 (the same issue of $K = F$ still exists and more experiments with larger K will be conducted).

The activation from matrix $\mathbf{Z}$ for the top 5 most frequently used dictionary items (with higher values of π_k) is shown in Figure 5. Note that the activation is sampled every 10 frames for better visualization. The reason that only the top 5 is chosen is that all the remaining items have significantly smaller π_k comparing with these 5. As we can see, there exists structure regarding different dictionary items. And even sparser structure can be expected with larger K. A non-extensive investigation shows that some of the dense regions in the activation corresponds to the appearance of particular speakers.

5 Conclusion

In this project, I extend the nonparametric Bayesian dictionary learning to incorporate temporal information for sound modeling. Preliminary experimental results show improvement comparing with the non-time-dependent model. Most of the future work has been mentioned above:

1. Add positive constrain to the dictionary and (probably) also to the weights, so that the reconstruction spectrogram $\mathbf{D}(\mathbf{S} \odot \mathbf{Z})$ does not contain non-positive elements.
2. Experiments with larger K are needed. Currently all the experiments are done with $K = 256$, partially due to the computational intensity (as explained below) of the model.
3. Add the coarse scale information into the model assumption as mentioned in Section 2.2.

As for the computational issue, one of the drawback with the current model is that it is built upon the dependency of the adjacent frames. Thus the Gibbs updating for $\boldsymbol{s}_i$ cannot be vectorized, as in order to update $\boldsymbol{s}_i$ we have to update $\boldsymbol{s}_{i-1}$ first, which significantly affects the overall efficiency. The complexity will grow linearly with the length of the input signal. Variational inference could be an alternative to look into as it generally much faster than sampling based algorithm. On the other hand, many of the Gibbs updatings can be done in parallel with different $k \in \{1, \cdots, K\}$. As K gets larger, this could be a huge gain.

References

[1] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Trans. Img. Proc.*, 15(12):3736–3745, December 2006.

[2] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun. Unsupervised learning of sparse features for scalable audio classification. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[3] Y. Karklin, C. Ekanadham, and E. P. Simoncelli. Hierarchical spike coding of sound. *Advances in Neural Information Processing Systems*, 25:3041–3049, 2012.

[4] H. Lee, Y. Largman, P. Pham, and A.Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Advances in neural information processing systems*, 22:1096–1104, 2009.

[5] J. Paisley and L. Carin. Nonparametric factor analysis with beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 777–784, 2009.

[6] J. Paisley, L. Carin, and D. Blei. Variational inference for stick-breaking beta process priors. In *Proceedings of the 28th Annual International Conference on Machine Learning*, 2011.

[7] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric bayesian dictionary learning for sparse image representations. *Advances in Neural Information Processing Systems*, 22:2295–2303, 2009.

A Complete Joint Posterior Distribution

In order to perform Gibbs sampling, we need to write down the complete posterior distribution. Assume our input data is a $F \times N$ matrix $\mathbf{X}$, the complete posterior distribution can be written as:

$$
\begin{aligned}
p(\Theta|\mathbf{X}) &\propto p(\mathbf{X}|\Theta)p(\Theta) \\
&= \prod_{i=1}^{N}\prod_{k=1}^{K} p(\boldsymbol{z}_i|\pi_k)p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{d}_k, \boldsymbol{s}_i, \gamma_\epsilon)p(\pi_k|a_0, b_0, K)p(\boldsymbol{d}_k|F)p(\boldsymbol{s}_i|\boldsymbol{s}_{i-1}, \gamma_s) \\
&\quad p(\gamma_s|c_0, d_0)p(\gamma_\epsilon|e_0, f_0) \\
&= \prod_{i=1}^{N}\prod_{k=1}^{K} \text{Bernoulli}(z_{ik}; \pi_k)\prod_{k=1}^{K}\text{Beta}(\pi_k; \frac{a_0}{K}, \frac{b_0(K-1)}{K})\mathcal{N}(\boldsymbol{d}_k; 0, F^{-1}\mathbf{I}_F) \\
&\quad \prod_{i=1}^{N}\mathcal{N}(\boldsymbol{x}_i; \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i), \gamma_\epsilon^{-1}\mathbf{I}_F) \cdot \mathcal{N}(\boldsymbol{s}_i; \boldsymbol{s}_{i-1}, \gamma_s^{-1}\mathbf{I}_K) \\
&\quad \Gamma(\gamma_s; c_0, d_0)\Gamma(\gamma_\epsilon; e_0, f_0)
\end{aligned}
$$

where $\Theta = \{\mathbf{D}, \mathbf{S}, \mathbf{Z}, K, \boldsymbol{\pi}, \gamma_s, \gamma_\epsilon, a_0, b_0, c_0, d_0, e_0, f_0\}$

B Gibbs Sampler Update

B.1 Samples $\boldsymbol{d}_k$

$$
\begin{aligned}
p(\boldsymbol{d}_k|-) &\propto \prod_{i=1}^{N}\mathcal{N}(\boldsymbol{x}_i; \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i), \gamma_\epsilon^{-1}\mathbf{I}_F) \cdot \mathcal{N}(\boldsymbol{d}_k; 0, F^{-1}\mathbf{I}_F) \\
&\propto \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{d}_k}, \boldsymbol{\Sigma}_{\boldsymbol{d}_k})
\end{aligned}
$$

where

$$\boldsymbol{\Sigma}_{\boldsymbol{d}_k} = (F\mathbf{I}_F + \gamma_\epsilon \sum_{i=1}^{N} z_{ik}^2 s_{ik}^2 \mathbf{I}_F)^{-1}$$

$$\boldsymbol{\mu}_{\boldsymbol{d}_k} = \gamma_\epsilon \boldsymbol{\Sigma}_{\boldsymbol{d}_k}\sum_{i=1}^{N} z_{ik}s_{ik}\tilde{\boldsymbol{x}}_i^{-k}$$

with

$$\tilde{\boldsymbol{x}}_i^{-k} = \boldsymbol{x}_i - \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i) + \boldsymbol{d}_k(\boldsymbol{s}_{ik} \odot \boldsymbol{z}_{ik})$$

B.2 Sample $\boldsymbol{z}_k = [z_{1k}, z_{2k}, \cdots, z_{Nk}]$

$$p(z_{ik}|-) \propto \text{Bernoulli}(z_{ik}; \pi_k) \cdot \mathcal{N}(\boldsymbol{x}_i; \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i), \gamma_\epsilon^{-1}\mathbf{I}_F)$$

$p(z_{ik} = 1|-)$ is proportional to

$$p_1 = \pi_k \exp[-\frac{\gamma_\epsilon}{2}(s_{ik}^2 \boldsymbol{d}_k^T\boldsymbol{d}_k - 2s_{ik}\boldsymbol{d}_k^T\tilde{\boldsymbol{x}}_i^{-k})]$$

$p(z_{ik} = 0|-)$ is proportional to

$$p_0 = 1 - \pi_k$$

Thus,

$$z_{ik} \sim \text{Bernoulli}(\frac{p_1}{p_0 + p_1})$$

B.3 Sample $\boldsymbol{s}_k = [s_{1k}, s_{2k}, \cdots, s_{Nk}]$

$$\begin{aligned} p(s_{ik}|-) &\propto \mathcal{N}(\boldsymbol{s}_i; \boldsymbol{s}_{i-1}, {\gamma_s}^{-1}\mathbf{I}_K)\mathcal{N}(\boldsymbol{s}_{i+1}; \boldsymbol{s}_i, {\gamma_s}^{-1}\mathbf{I}_K)\mathcal{N}(\boldsymbol{x}_i; \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i), {\gamma_\epsilon}^{-1}\mathbf{I}_F) \\ &\propto \mathcal{N}(\mu_{s_{ik}}, \Sigma_{s_{ik}}) \end{aligned}$$

where

$$\begin{aligned} \Sigma_{s_{ik}} &= (2\gamma_s + \gamma_\epsilon z_{ik}^2 \boldsymbol{d}_k^T \boldsymbol{d}_k)^{-1} \\ \mu_{s_{ik}} &= \Sigma_{s_{ik}}[\gamma_s(s_{(i-1)k} + s_{(i+1)k}) + \gamma_\epsilon z_{ik} \boldsymbol{x}_i^T \boldsymbol{d}_k] \end{aligned}$$

B.4 Sample π_k

$$\begin{aligned} p(\pi_k|-) &\propto \text{Beta}(\pi_k; \frac{a_0}{K}, \frac{b_0(K-1)}{K}) \prod_{i=1}^{N} \text{Bernoulli}(z_{ik}; \pi_k) \\ &\propto \text{Beta}(\frac{a_0}{K} + \sum_{i=1}^{N} z_{ik}, \frac{b_0(K-1)}{K} + N - \sum_{i=1}^{N} z_{ik}) \end{aligned}$$

B.5 Sample γ_s

$$\begin{aligned} p(\gamma_s|-) &\propto \Gamma(\gamma_s; c_0, d_0) \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{s}_i; \boldsymbol{s}_{i-1}, {\gamma_s}^{-1}\mathbf{I}_K) \\ &\propto \Gamma\Big(c_0 + \frac{1}{2}KN, d_0 + \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{s}_i - \boldsymbol{s}_{i-1})^T(\boldsymbol{s}_i - \boldsymbol{s}_{i-1})\Big) \end{aligned}$$

B.6 Sample γ_ϵ

$$\begin{aligned} p(\gamma_\epsilon|-) &\propto \Gamma(\gamma_\epsilon; e_0, f_0) \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{x}_i; \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i), {\gamma_s}^{-1}\mathbf{I}_F) \\ &\propto \Gamma\Big(e_0 + \frac{1}{2}FN, f_0 + \frac{1}{2}\sum_{i=1}^{N} \big(\boldsymbol{x}_i - \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i)\big)^T \big(\boldsymbol{x}_i - \mathbf{D}(\boldsymbol{s}_i \odot \boldsymbol{z}_i)\big)\Big) \end{aligned}$$