Kruskal-Factor Analysis
Supplementary Material

1 Gibbs updates

KFA Likelihood and priors:

$$\begin{aligned}
\mathcal{X}_n &\sim \mathcal{N}(\mathcal{D} \underset{T+1}{\times} (\boldsymbol{s}_n \circ \boldsymbol{z}_n), \gamma_\epsilon^{-1}\boldsymbol{I}),\ d_{\mathsf{i}k} = \sum_{r=1}^{R} \lambda_{rk} \prod_{t=1}^{T} u^{(kt)}_{\mathsf{i}_t r} \\
u^{(kt)}_{\mathsf{i}_t r} &\sim \mathcal{N}(0, m_t^{-1}),\ \lambda_{kr} \sim \mathcal{N}(0, \tau_{kr}^{-1}),\ \tau_{kr} = \prod_{i=1}^{r} \delta_{ki} \\
s_{kn} &\sim \mathcal{N}(0, \gamma_s^{-1}), \quad z_{kn} \sim \mathsf{Bernoulli}(\pi_k) \\
\pi_k &\sim \mathsf{Beta}\left(a_\pi/K,\ b_\pi(K-1)/K\right), \quad \delta_{ki} \sim \mathsf{Gam}(\alpha, 1)
\end{aligned} \tag{1}$$

where $\mathcal{X}_n \in \mathbb{R}^{m_1 \times \cdots \times m_T}$ is the n^{th} data item, and $d_{\mathsf{i}k}$ is the $\mathsf{i} = [\mathsf{i}_1, \ldots, \mathsf{i}_T]$ element of the k^{th} atom. The precisions $\gamma_\epsilon, \gamma_s$ have gamma priors.

$$\begin{aligned}
d_{\mathsf{i}k} &= \lambda_{rk} \prod_{t' \neq t} u^{(kt')}_{\mathsf{i}_{t'} r} u^{(kt)}_{\mathsf{i}_t r} + \sum_{r' \neq r} \lambda_{r'k} \prod_{t=1}^{T} u^{(kt)}_{\mathsf{i}_t r'} = a_{\mathsf{i}_t} u^{(kt)}_{\mathsf{i}_t r} + b_{\mathsf{i}_t} \\
&= \prod_{t=1}^{T} u^{(kt)}_{\mathsf{i}_t r} \lambda_{rk} + \sum_{r' \neq r} \lambda_{r'k} \prod_{t=1}^{T} u^{(kt)}_{\mathsf{i}_t r'} \quad = f_{\mathsf{i}} \lambda_{rk} + g_{\mathsf{i}}
\end{aligned} \tag{2}$$

$$\begin{aligned}
u^{(kt)}_{\mathsf{i}_t r} &\sim \mathcal{N}(\hat{\mu}, \hat{\omega}^{-1}) \\
\hat{\omega} &= m_t + \sum_{\mathsf{i}:\mathsf{i}_t = m} \gamma^{\mathcal{D}_k}_{\mathsf{i}_t k} \left(a_{\mathsf{i}_t}\right)^2 \\
\hat{\mu} &= \frac{1}{\hat{\omega}} \sum_{\mathsf{i}:\mathsf{i}_t = m} \left(\mu^{\mathcal{D}_k}_{\mathsf{i}_t k} - b_{\mathsf{i}_t}\right) a_{\mathsf{i}_t} \\
\gamma^{\mathcal{D}_k} &= \gamma_\epsilon \boldsymbol{I}_P \sum_{n=1}^{N} s^2_{kn} z^2_{kn} \\
\mu^{\mathcal{D}_k} &= \gamma_\epsilon \sum_{n=1}^{N} s_{kn} z_{kn} \tilde{\boldsymbol{x}}_n^{\setminus k} \\
\tilde{\boldsymbol{x}}_n^{\setminus k} &= \text{vec}\left[\mathcal{X}_n - \mathcal{D} \underset{T+1}{\times} (\boldsymbol{s}_n \circ \boldsymbol{z}_n) + s_{kn} z_{kn} \mathcal{D}_k\right]
\end{aligned} \tag{3}$$

$$
\begin{aligned}
\lambda_{kr} &\sim \mathcal{N}(\hat{\nu}, \hat{\tau}^{-1}) \\
\hat{\tau} &= \tau_{kr} + \gamma_\epsilon \sum_{n,\mathrm{i}} \left(f_\mathrm{i} s_{kn}\right)^2 \\
\hat{\nu} &= \frac{\gamma_\epsilon}{\hat{\tau}} \sum_{n,\mathrm{i}} \left(h_{\mathrm{i}n} - g_\mathrm{i} s_{kn}\right) f_\mathrm{i} s_{kn} \\
h_{\mathrm{i}n} &= \mathcal{X}_{\mathrm{i}n} - [\mathcal{D} \underset{T+1}{\times} (\boldsymbol{s}_n \circ \boldsymbol{z}_n)]_\mathrm{i}
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
p(s_{ki}|-) &\sim \mathcal{N}(\mu_{s_{ki}}, \Sigma_{s_{ki}}) \\
\Sigma_{s_{ki}} &= (\gamma_s + \gamma_\epsilon z_{kn}^2 \boldsymbol{d}_k^T \boldsymbol{d}_k)^{-1} \\
\mu_{s_{ki}} &= \gamma_\epsilon z_{kn} \boldsymbol{d}_k^T \tilde{\boldsymbol{x}}_n^{\setminus k} \\
\boldsymbol{d}_k &= \mathrm{vec}[\mathcal{D}_k]
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
z_{ki} &\sim \mathrm{Bern}\left(\frac{p_1}{p_0 + p_1}\right) \\
p_1 &= \pi_k \exp[-\frac{\gamma_\epsilon}{2}(s_{kn}^2 \boldsymbol{d}_k^T \boldsymbol{d}_k - s_{kn} \boldsymbol{d}_k^T \tilde{\boldsymbol{x}}_n^{\setminus k})] \\
p_0 &= 1 - \pi_k
\end{aligned}
\tag{6}
$$

$$
p(\gamma_\epsilon|-) \sim \Gamma\left(c + \frac{PN}{2},\, d + \frac{1}{2}\sum_{n=1}^{N} \|\mathcal{X}_n - \mathcal{D} \underset{T+1}{\times} (\boldsymbol{s}_n \circ \boldsymbol{z}_n))\|_F^2\right) \tag{7}
$$

$$
p(\gamma_s|-) \sim \Gamma\left(e + \frac{KN}{2},\, f + \frac{1}{2}\sum_{n=1}^{N} \boldsymbol{s}_n^T \boldsymbol{s}_n\right) \tag{8}
$$

$$
p(\pi_k|-) \sim \mathrm{Beta}\left(\frac{a}{K} + \sum_{n=1}^{N} z_{kn},\, b\frac{K-1}{K} + N - \sum_{n=1}^{N} z_{kn}\right) \tag{9}
$$

2 Bayesian Conditional Density Filter for KFA

We record surrogate conditional sufficient statistics (SCSS) for the global parameters that directly interact with the data or local parameters $\mathcal{D}, \gamma_\epsilon, \gamma_s, \pi_k$. The SCSS are given by the posterior updates of the global parameters. The SCSS are accumulated for each subset of data, and then applied in the subsequent update as the prior parameters. We reset the SCSS after the number of data items seen is the same as the size of the dataset. In our implementation we update the model parameters 3 times using a single subset of data. The SCSS are held fixed until the 3$^{\text{rd}}$ iteration. This allows stale local parameters $\{\boldsymbol{S}, \boldsymbol{Z}\}$ to be refreshed before updating the SCSS. The algorithm below describes the process.

for *epoch* $= 1$ **to** *numEpoch* **do**
  Randomly partition $\mathcal{X}$ into minibatches $\mathcal{X}^{(i)}$ of size *batchSize*
  Set SCSS to prior parameters
  for all $\mathcal{X}^{(i)}$ **do**
    Update $\boldsymbol{S}^{(i)}, \boldsymbol{Z}^{(i)}$ and all hyperparameters with KFA
    Do a standard KFA update twice (including the dictionary)
    Update SCSS ($\forall k$):

$$\boldsymbol{\mu}^{\mathcal{D}_k}_{\text{SCSS}} = \boldsymbol{\mu}^{\mathcal{D}_k}_{\text{SCSS}} + \boldsymbol{\mu}^{\mathcal{D}_k(i)}$$
$$\boldsymbol{\gamma}^{\mathcal{D}_k}_{\text{SCSS}} = \boldsymbol{\gamma}^{\mathcal{D}_k}_{\text{SCSS}} + \boldsymbol{\gamma}^{\mathcal{D}_k(i)}$$
$$\pi_{ka\text{SCSS}} = \pi_{ka\text{SCSS}} + \textstyle\sum_n z^{(i)}_{kn}$$
$$\pi_{kb\text{SCSS}} = \pi_{kb\text{SCSS}} + N - \textstyle\sum_n z^{(i)}_{kn}$$
$$\gamma_{sa\text{SCSS}} = \gamma_{sa\text{SCSS}} + KN^{(i)}/2$$
$$\gamma_{sb\text{SCSS}} = \gamma_{sb\text{SCSS}} + \tfrac{1}{2}\textstyle\sum_n \boldsymbol{s}^{(i)\top}_n \boldsymbol{s}^{(i)}_n$$
$$\gamma_{\epsilon a\text{SCSS}} = \gamma_{\epsilon a\text{SCSS}} + PN^{(i)}/2$$
$$\gamma_{\epsilon b\text{SCSS}} = \gamma_{\epsilon b\text{SCSS}} + \tfrac{1}{2}\textstyle\sum_n \|\mathcal{X}^{(i)}_n - \mathcal{D} \underset{T+1}{\times} (\boldsymbol{s}^{(i)}_n \circ \boldsymbol{z}^{(i)}_n)\|^2_F$$

  end for
  Reset SCSS to original prior parameters
  if *epoch* $== 1$ **then**
    Update $\boldsymbol{S}, \boldsymbol{Z}$ and all hyperparameters using the full data set (note that every column is independent, so this can be done in parallel).
  end if
end for

3 Barbara Images

Figure 1: Barbara reconstructions from a random 20% of the pixels: corrupted image (top), KFA (center), BPFA (bottom). KFA reconstructs the eyes and mouth better than BPFA. Also, the contrast in the stripe pattern is stronger in the KFA reconstruction.

4 Dictionary Recovery

For this example, we assume the dictionary is composed of the $5 \times 5 \times 5 \times 5$ DCT filters. Note that each filter is rank-1 (separable). We generate 500 data samples using normal random weights for $\boldsymbol{S}$, enforce that only 8 nonzero elements are present in each column of $\boldsymbol{Z}$ and set the noise variance to 0.01. We use the same settings for BPFA and KFA, with $R = 1$. The dictionary size is $K = 625$, the same as the number of DCT filters. We ran both models for 100 iterations. Figure 2 shows the learned dictionaries from each algorithm. KFA learns the DCT filters. We find that even when we set R to the expected rank $R_E = 30$, the KFA dictionary atoms are low-rank. The BPFA dictionary looks noisy (high-rank structures) while both of the KFA dictionaries have low-rank atoms.

The RMSE for the three models is given in table 1. We can see from the dictionary and the RMSE that KFA is discovering the multi-way structure of the data. The number of samples is smaller than the dictionary, yet KFA is still able to find a well structured dictionary and achieve a lower error.

Table 1: RMSE on 4D-DCT example

	BPFA	KFA $R = 1$	KFA $R = 30$
RMSE	0.03795	0.01387	0.01095

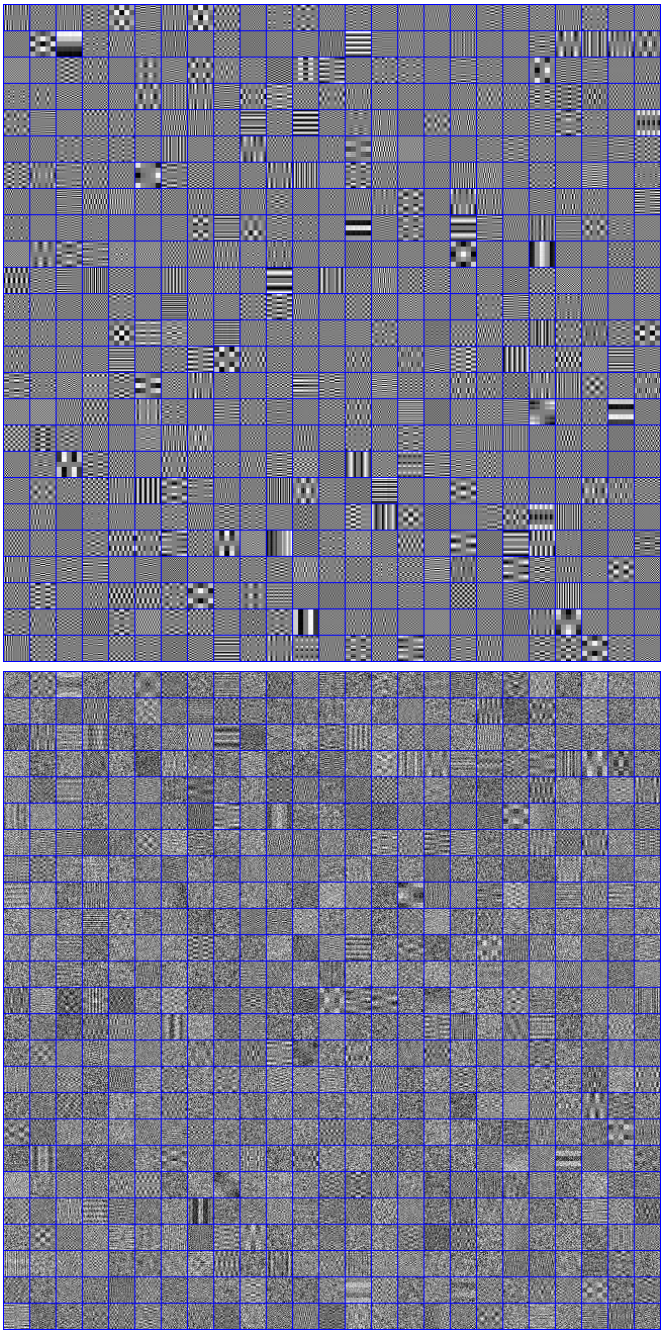

Figure 2: 4D-DCT example dictionaries collapsed to 2D: KFA (top), BPFA (bottom). Zoom-in for fine structure.