COMPARING HYPERPARAMETERS

V. CHEN

1. Comparing v versus τ, α. Here, we compare models (C) and (E). Recall that in model (C), for the noncentered approach we use the map given by

$$T(\xi, \tau, \alpha, M) = \sum_{j=0}^{M} (\lambda_j + \tau^2)^{-\alpha/2} \xi_j q_j.$$

We impose uniform priors over intervals on τ, α, M. One change made to this algorithm is to scale $T(\xi, \theta)$ so that $\mathbb{E}(u_j^2) = N$ in the prior for u. This can be done by scaling $T \to \sqrt{\frac{N}{\text{Tr}((L+\tau^2)^{-\alpha})}} T$.

In model (E), we use

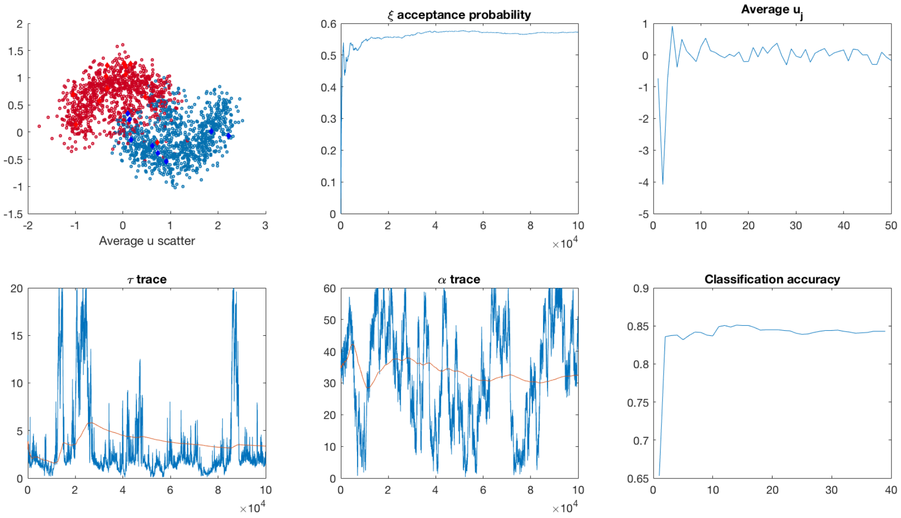
$$T(\xi, v, M) = \sum_{j=0}^{M} v_j \xi_j q_j$$

with $v_j \sim \mathsf{U}\left((1-a)(\lambda_j + \tau^2)^{-\alpha/2}, (1+a)(\lambda_j + \tau^2)^{-\alpha/2}\right)$. We compare the performance of these two models on the two moons dataset and on MNIST.

2. Two moons. With $\sigma = 0.2$, 1% fidelity, $r = 1$, $d = 100$, and $N = 2000$, we generate realizations of two moons.

2.1. Model (C). We first run model (C) with fixed $M = 50$ on this dataset.

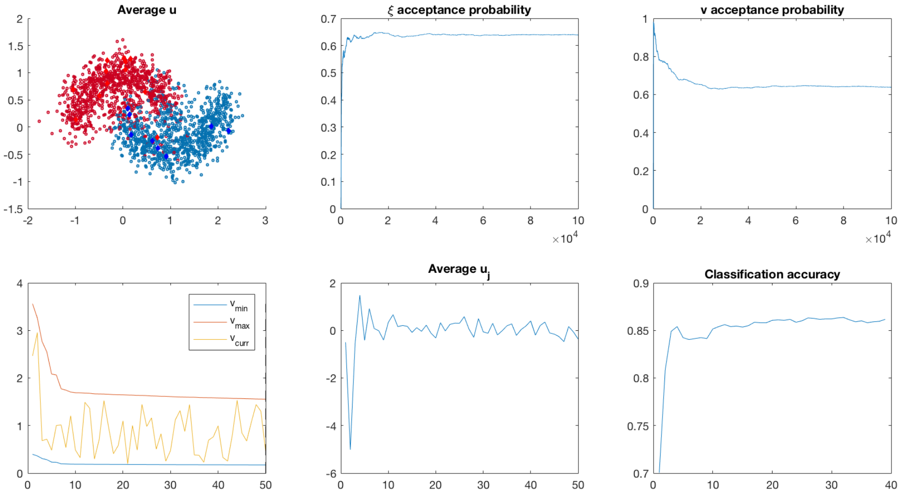
FIG. 1. *Model (C) on two moons. Figures from left to right, top to bottom: Final classification obtained, ξ running acceptance probability, final average of u_j, τ trace, α trace, running classification accuracy (updated every 2500 trials).*



As shown in Figure 1, this algorithm achieves around 85% classification accuracy and the MCMC converges in both acceptance probability and classification accuracy after around 10000 iterations.

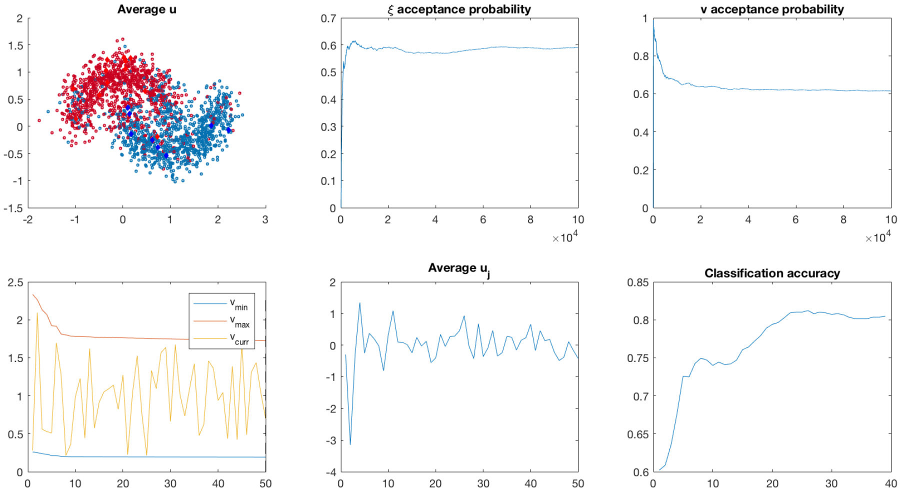
2.2. Model (E). If we choose the parameters of the prior on v_j to be $\tau = 3, \alpha = 35, a = 0.8$ and fix $M = 50$ for model (E), we can obtain levels of accuracy and convergence rates similar to model (C). Note that this is "cheating" in the sense that it uses the convergence of τ in model (C). The results are shown in Figure 2.

Fig. 2. *Model (E) on two moons with $\tau = 3, \alpha = 35, a = 0.8$. Figures from left to right, top to bottom: Final classification obtained, ξ running acceptance probability, v acceptance probability, final v_j observation, final average of u_j, running classification accuracy (updated every 2500 trials).*



The same accuracy is not achieved when $\tau = 5$ is chosen. See Figure 3. Note that convergence of the classification accuracy appears much slower and the final accuracy is still lower than the previous two examples.

Fig. 3. *Model (E) on two moons with $\tau = 5, \alpha = 35, a = 0.8$.*

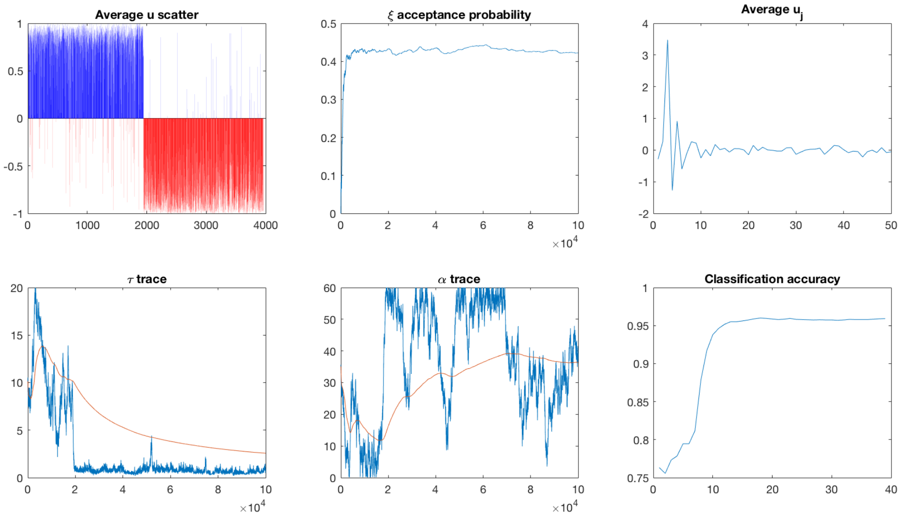


It seems that model (E) is very sensitive to the value of τ chosen for its prior. Initializing τ to be at the value suggested by model (C) achieves similar results in both classification accuracy and convergence rate, but choosing a somewhat poor value of τ leads to a noticeable drop in accuracy and convergence rate.

3. MNIST. We compare these two algorithms on MNIST binary classification of 4 and 9. We set $\gamma = 0.0001$ for the label noise (this seems to improve accuracy for both models).

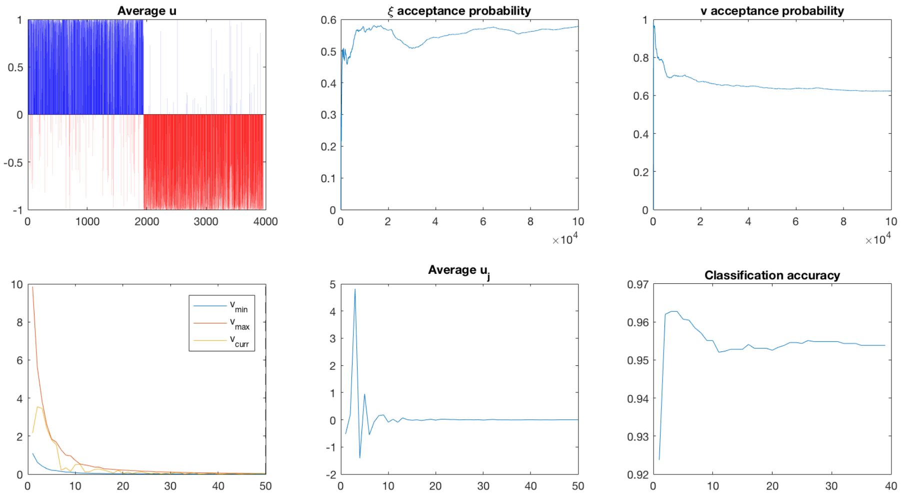
3.1. Model (C). We fix $M = 50$ and allow τ, α to be learnt from uniform priors $[0.01, 20]$ and $[0.1, 60]$, respectively. The results are shown in Figure 4. τ is initialized at 20 but finds a small value at around step 20000 and stays around there. Note that this corresponds with the sharp increase in classification accuracy after this value of τ was found. The mean and median of τ after it seems to converge is around 0.7. The accuracy is around 96%.

FIG. 4. *Model (C) on MNIST49. Figures from top to bottom, left to right: Final classification obtained, ξ running acceptance probability, final average of u_j, τ trace, α trace, running classification accuracy (updated every 2500 trials).*



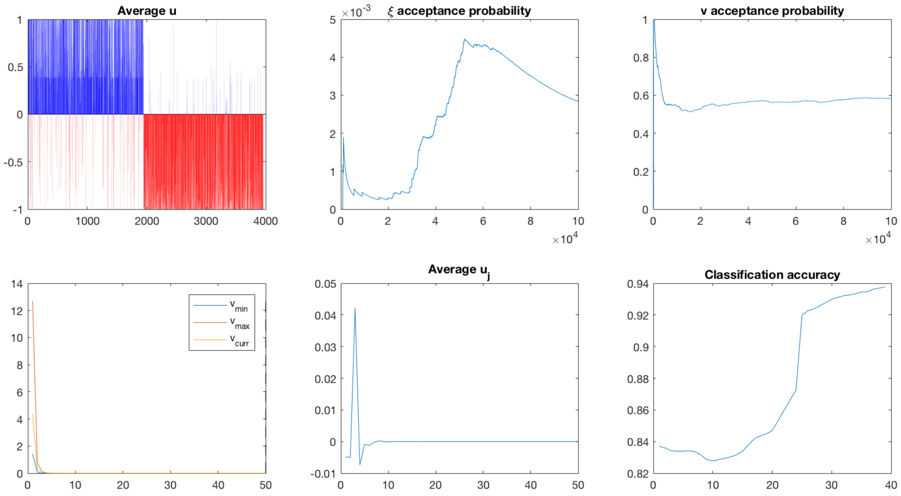
3.2. Model (E). Fix $M = 50$ again. With $\tau = 0.7$ (which is "cheating" by using the τ learnt from model (C)), we obtain Figure 5. Note the similar final accuracy of around 96%, with faster convergence to that accuracy since we cheated with the initialization.

FIG. 5. *Model (E) on two moons with $\tau = 0.7, \alpha = 35, a = 0.8$. Figures from left to right, top to bottom: Final classification obtained, ξ running acceptance probability, v acceptance probability, final v_j observation, final average of u_j, running classification accuracy (updated every 2500 trials).*



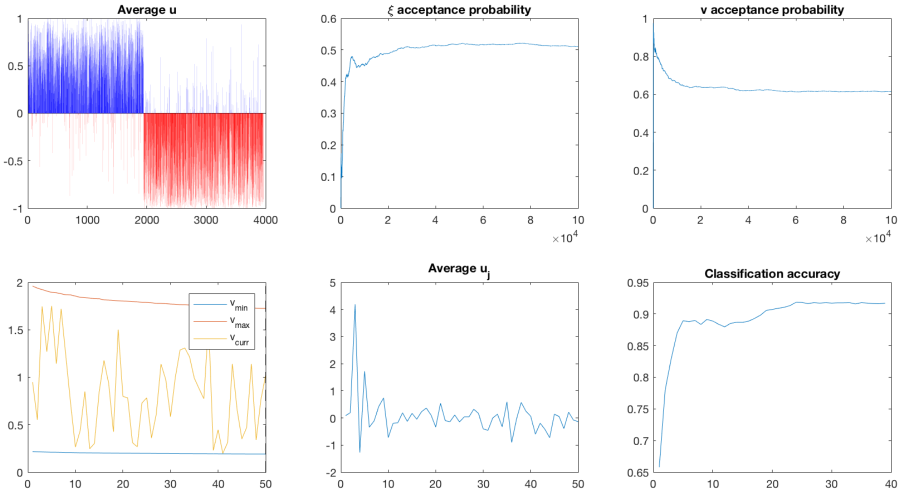
With $\tau = 0.3$, we obtain Figure 6. Note the slow convergence compared to $\tau = 0.7$. In fact, it appears that the accuracy is still climbing even at 100000 iterations.

FIG. 6. *Model (E) on two moons with $\tau = 0.3, \alpha = 35, a = 0.8$.*



With $\tau = 5$, we obtain Figure 7. Note the overall lower final accuracy of around 90%.

FIG. 7. *Model (E) on two moons with $\tau = 5, \alpha = 35, a = 0.8$.*



4. Observations on τ, α.

This section attempts to summarize the discussions on the effects of τ and α.

With small fixed values of α, such as $\alpha = 1$, the variance of the Gaussian prior on u_j decreases at a slower rate with increasing values of j. This means that the algorithm is more able to draw samples that use eigenvectors with a large range of indices. If the problem can be solved with a small number of eigenvectors, we expect α to be larger.

The smallest eigenvectors of the graph Laplacian should behave as indicators of the clusters and have eigenvalues close to zero. This means $\lambda + \tau^2$ will appear to be close to τ^2 for these eigenvectors. τ should be large enough so that the eigenvectors needed have similar values of

$\lambda + \tau^2$, but must be small enough so that the unnecessary eigenvectors do not also appear to have the same value of $\lambda + \tau^2$.

REFERENCES