WEEK 2 SUMMARY

V. CHEN

1. Summary. Last week's experiments indicated that the original parameterization, treating u, τ, α as the variables, would not converge even after 100,000 iterations. First, we explain the different parameterization that we tried this week. We decided to try a different parameterization which treats ξ, τ, α as variables to sample, and we compare results between this and the old parameterization.

2. Learning τ and α with non-centered parameterization.

2.1. Algorithm. In the centered parameterization, the prior was $u|\tau, \alpha \sim N(0, C(\tau, \alpha))$, τ, α distributed independently and uniformly over intervals. We would take the posterior by Bayes' theorem to be

$$f(u, \tau, \alpha) \propto \exp\left(-\Phi(u) - \frac{1}{2}\langle u, C(\tau, \alpha)^{-1}u\rangle - \frac{1}{2}\log(\det(C(\tau, \alpha))) + \log(\pi_0(\tau, \alpha))\right)$$

In the new parameterization, we sample $\xi \sim N(0, I)$ and τ, α on uniform intervals as before. ξ is related to u by $T(\xi, \tau, \alpha) = u = \sum_{j=0}^{M}(\lambda_j + \tau^2)^{-\alpha/2}\xi_j q_j$. Recall that we are taking $M = N - 1$. The new joint posterior becomes:

$$g(\xi, \tau, \alpha) \propto \exp\left(-\Phi(T(\xi, \tau, \alpha)) - \frac{1}{2}\langle u, u\rangle + \log(\pi_0(\tau, \alpha))\right)$$

This algorithm works as follows:

Algorithm 1 Non-centered parameterization: sampling ξ, τ, α

Choose $\xi^{(0)} \in \mathbb{R}^N, \alpha^{(0)}, \tau^{(0)} > 0, \beta \in (0, 1]$ and $\epsilon_1, \epsilon_2 > 0$.
for $k = 0$ to S **do**
Propose $\hat{\xi}^{(k)} = (1 - \beta^2)\xi^{(k)} + \beta\zeta^{(k)}$, $\zeta^{(k)} \sim N(0, I)$
Make transition $\xi^{(k+1)} \to \hat{\xi}^{(k)}$ with probability

$$A(\xi^{(k)} \to \hat{\xi}^{(k)}) = \min\{1, \frac{g(\hat{\xi}^{(k)}, \tau^{(k)}, \alpha^{(k)})}{g(\xi^{(k)}, \tau^{(k)}, \alpha^{(k)})}\}$$

Propose $\hat{\tau}^{(k)} = \tau^{(k)} + \epsilon_1\rho^{(k)}, \rho^{(k)} \sim N(0, I)$
Make transition $\tau^{(k+1)} \to \hat{\tau}^{(k)}$ with probability

$$A(\tau^{(k)} \to \hat{\tau}^{(k)}) = \min\{1, \frac{g(\xi^{(k+1)}, \hat{\tau}^{(k)}, \alpha^{(k)})}{g(\xi^{(k+1)}, \tau^{(k)}, \alpha^{(k)})}\}$$

Propose $\hat{\alpha}^{(k)} = \alpha^{(k)} + \epsilon_2\sigma^{(k)}, \sigma^{(k)} \sim N(0, I)$
Make transition $\alpha^{(k+1)} \to \hat{\alpha}^{(k)}$ with probability

$$A(\alpha^{(k)} \to \hat{\alpha}^{(k)}) = \min\{1, \frac{g(\xi^{(k+1)}, \tau^{(k+1)}, \hat{\alpha}^{(k)})}{g(\xi^{(k+1)}, \tau^{(k+1)}, \alpha^{(k)})}\}$$

end for
return $\{T(\xi^{(k)}, \tau^{(k)}, \alpha^{(k)}), \tau^{(k)}, \alpha^k\}$

2.2. Simulation results. The relevant figures are Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6. The parameters used are as follows:

Iterations	100000
Burn in period	1000
Laplacian	self-tuning, unnormalized
β	0.1
γ	0.0001
Labeled +1	$280 - 290$
Labeled -1	$20 - 30$
ϵ_α	1
ϵ_τ	1
Initial τ	30
Initial α	5
τ range	$[0, 60]$
α range	$[0, 100]$
Average τ	3.74
Average α	63.5
Percent of correct classification	0.864407
Time elapsed	183.21 s

FIG. 1. *α acceptance probability*

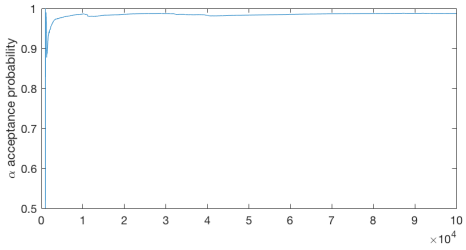


FIG. 2. *α trace*

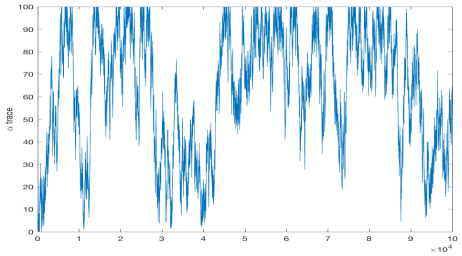


FIG. 3. *τ acceptance probability*

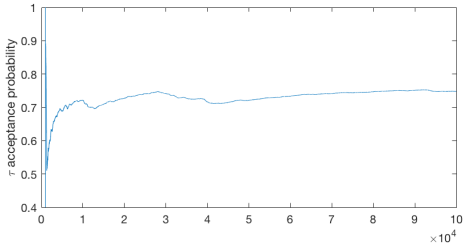


FIG. 4. *τ trace*

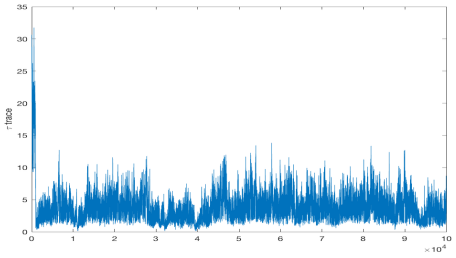


FIG. 5. *ξ acceptance probability*

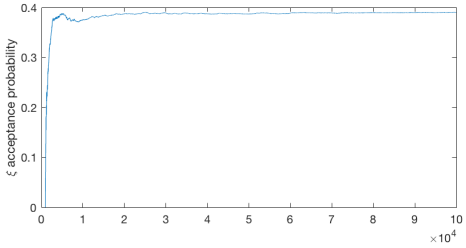
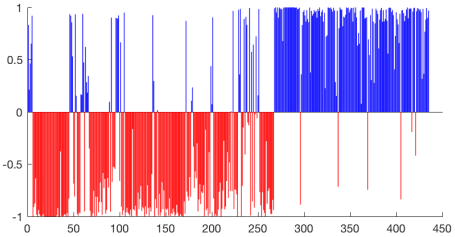


FIG. 6. *u final average*



3. Revisiting the centered parameterization. Matt took a look at my code and suggested that part of the reason why the parameters do not mix well is because of the accuracy of Matlab's eig solver for large eigenvalues, since the eigenvectors are very irregular. From Matt's suggestion, I truncated the sum after $M = 50$ terms, a spectral projection. Applying this algorithm to the voting data set, the average u was able to quickly converge to a function similar to the Fiedler vector, and the hyperparameters τ and α showed better mixing.

Laplacian	self-tuning, unnormalized
β	0.4
γ	0.0001
Labeled +1	$280 - 290$
Labeled -1	$20 - 30$
ϵ_α	0.1
ϵ_τ	1
Initial τ	20
Initial α	20
Percent of correct classification	0.837772

FIG. 7. *τ trace*

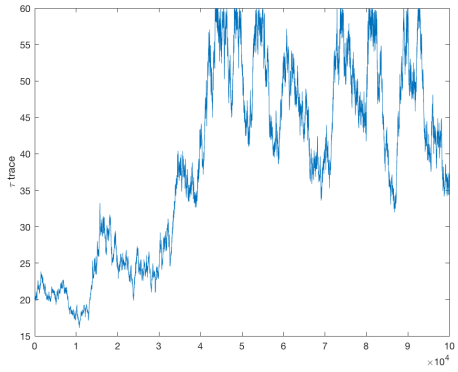
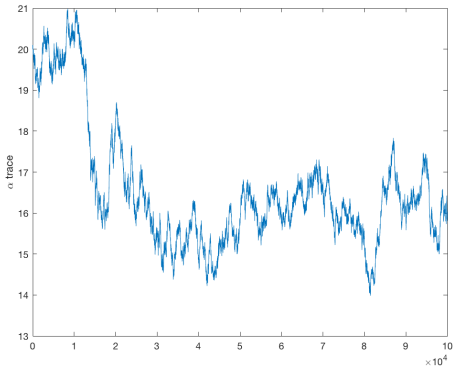


FIG. 8. *α trace*



4. Two moons data. We began to test our algorithms on the two moons data. We used 1000 nodes and noise in 100 dimensions with standard deviation $\sigma = 0.1$. We labeled about 4% of the nodes.

4.1. Centered parameterization. The centered parameterization with the unnormalized Laplacian and without truncation does not perform well on the two moons data set. This could again be due to higher order eigenvectors. See Figure 9, Figure 10, Figure 11, Figure 12.
Truncating to $M = 50$ eigenvectors greatly improved classification and mixing of the hyperparameters. See Figure 13, Figure 14, Figure 15, Figure 16.

4.2. Non-centered parameterization. Working with the non-centered parameterization, we see that spectral projection/truncating has similar effects when using the unnormalized Laplacian: the algorithm has difficulty clustering the two moons without truncation, and is able to cluster when truncating to $M = 50$ eigenvectors, with an accuracy of about 80%. See Figure 17, Figure 18, Figure 19, Figure 20.
However, we get better results with the non-centered parameterization when using the self-tuning unnormalized Laplacian. It seems that the self-tuning Laplacian offers more clustering information in its higher order eigenvectors, while the eigenvectors beyond the Fiedler vector in the unnormalized Laplacian are less useful. Refer to Figure 21, Figure 22. Clustering accuracy is around 98%. See Figure 23, Figure 24, Figure 25, Figure 26.

FIG. 9. *Without truncation, τ trace*

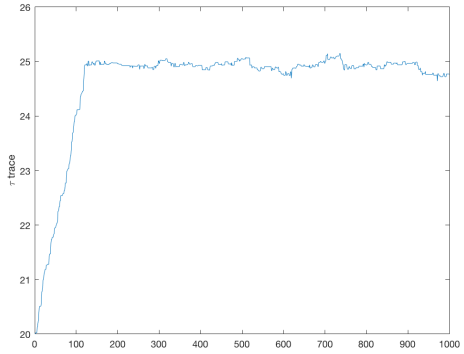


FIG. 10. *Without truncation, α trace*

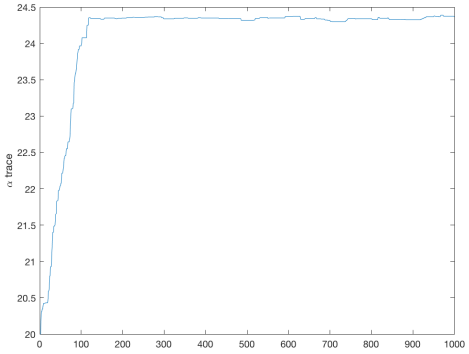


FIG. 11. *Without truncation, average eigenfunction u*

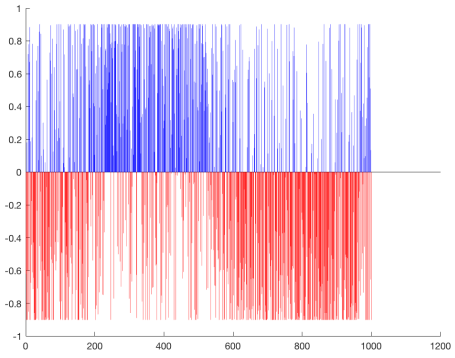


FIG. 12. *Without truncation, final clustering obtained. Colored diamonds indicate the labeled nodes.*

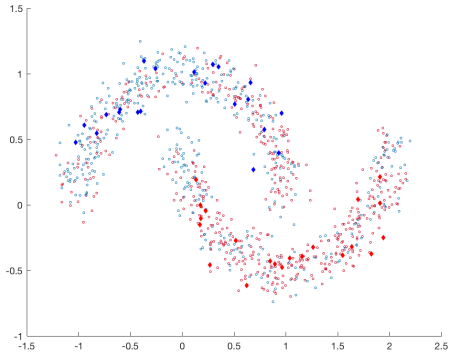


FIG. 13. *Truncated, τ trace*

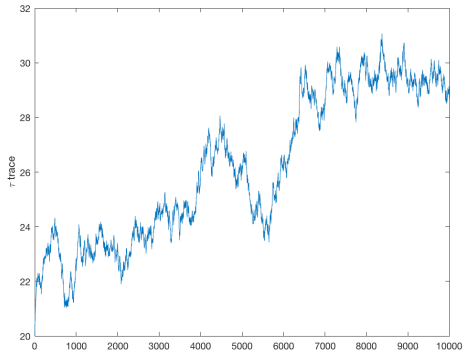


FIG. 14. *Truncated, α trace*

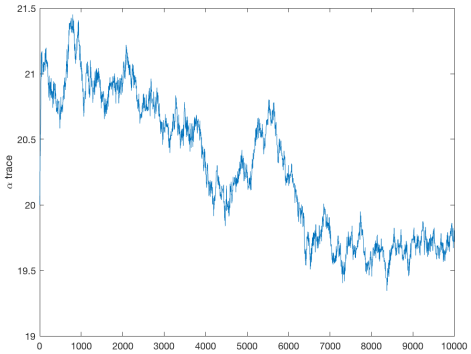


FIG. 15. *Truncated, average eigenfunction u*

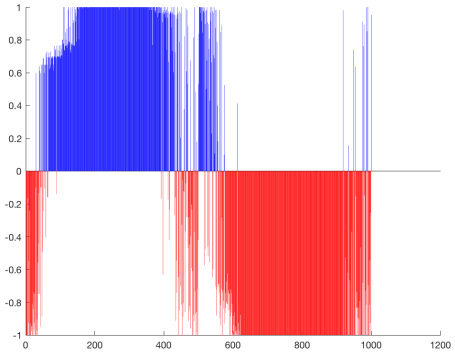


FIG. 16. *Truncated, final clustering obtained*

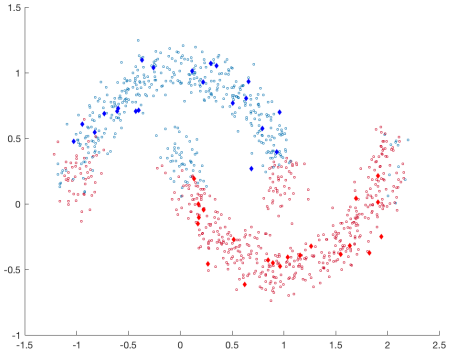


FIG. 17. *Noncentered truncated, τ trace*

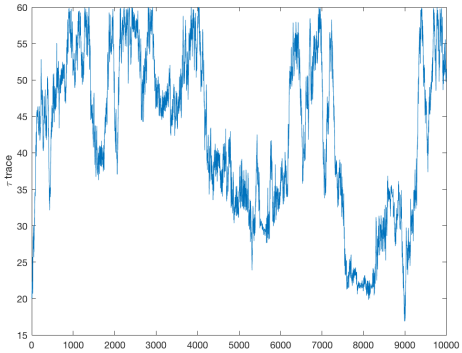


FIG. 18. *Noncentered truncated, α trace*

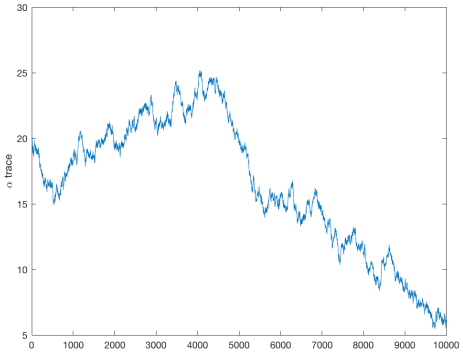


FIG. 19. *Noncentered truncated, average eigenfunction u*

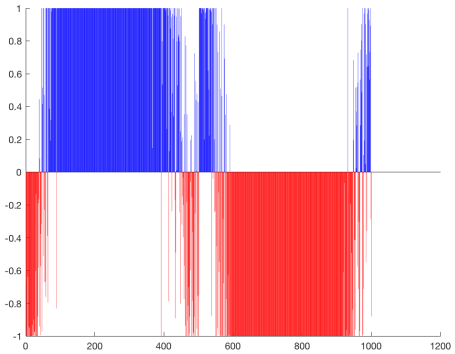


FIG. 20. *Noncentered truncated, final clustering obtained*

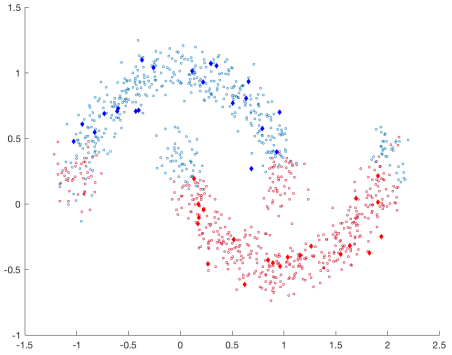


FIG. 21. *First eigenvectors of unnormalized Laplacian for intertwined moons data*

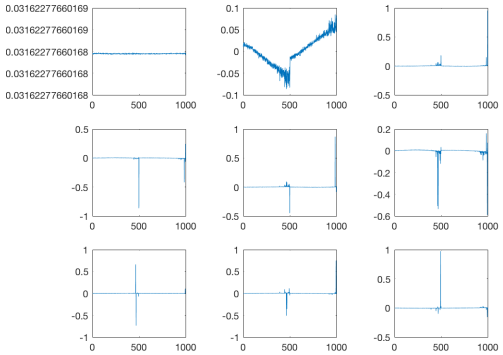


FIG. 22. *First eigenvectors of unnormalized self-tuning Laplacian for intertwined moons data*

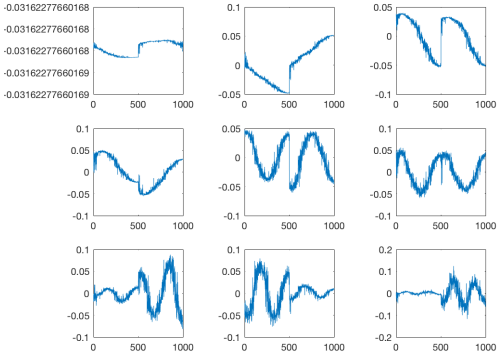


FIG. 23. *Self tuning, τ trace*

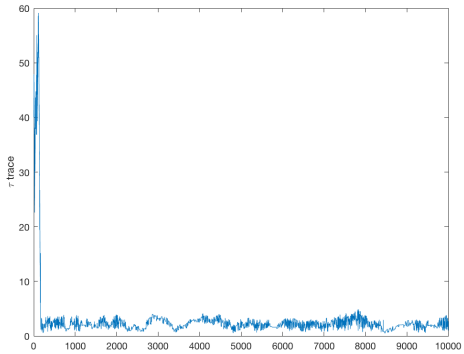


FIG. 24. *Self tuning, α trace*

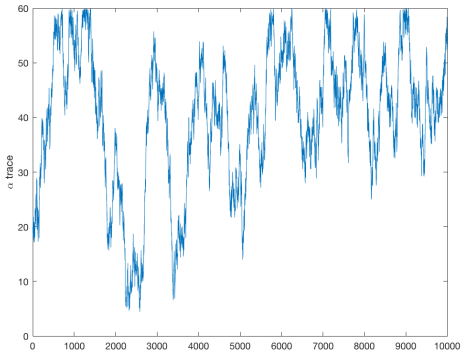


FIG. 25. *Self tuning, average eigenfunction u*

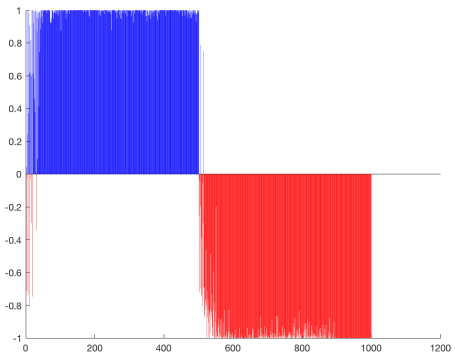


FIG. 26. *Self tuning, clustering obtained*