Appendix: Kronecker Determinantal Point Processes

A Proof of Prop. 3.1

We use 'vec' to denote the operator that stacks columns of a matrix to form a vector; conversely, 'mat' takes a vector with k^2 coefficients and returns a $k \times k$ matrix.

Let $L = L_1 \otimes L_2$, $S_1 = L_1^{-1}, S_2 = L_2^{-1}$ and $S = S_1 \otimes S_2 = L^{-1}$. Let E_{ij} be the matrix with all 0s except for a 1 at position (i, j), its size being clear from context. We wish to solve

$$\nabla f_2(X) = -\nabla g_2(S_1) \quad \text{and } \nabla f_1(X) = -\nabla g_1(S_2) \tag{10}$$

It follows from the fact that

$$\log\det(S_1 \otimes S_2) = N_2 \log\det S_1 + N_1 \log\det S_2$$

that $\nabla f_{S_2}(X) = N_2 X^{-1}$ and $\nabla f_{S_1}(X) = N_1 X^{-1}$. Moreover, we know that

$$\begin{aligned}
\nabla g(S) &= -(I+S)^{-1} - S^{-1}\frac{1}{n}\sum\nolimits_i U_i (U_i^\top S^{-1} U_i)^{-1} U_i S^{-1} \\
&= -S^{-1} - S^{-1}\left(\frac{1}{n}\sum\nolimits_i U_i (U_i^\top S^{-1} U_i)^{-1} U_i - (I + S^{-1})^{-1}\right) S^{-1} \\
&= -(L + L\Delta L).
\end{aligned}$$

The Jacobian of $S_1 \to S_1 \otimes S_2$ is given by $J = (\operatorname{vec}(E_{11} \otimes S_2), \dots, \operatorname{vec}(E_{N_1 N_1} \otimes S_2))$. Hence,

$$\begin{aligned}
\nabla f_1(X)_{ij} = -(\nabla g_1(S_1))_{ij} &\iff N_2 X_{ij}^{-1} = (J^\top \operatorname{vec}(-\nabla g(S)))_{ij} \\
&\iff N_2 X_{ij}^{-1} = \operatorname{vec}(E_{ij} \otimes S_2)^\top \operatorname{vec}(L + L\Delta L) \\
&\iff N_2 X_{ij}^{-1} = \operatorname{Tr}((E_{ij} \otimes S_2)(L + L\Delta L)) \\
&\iff N_2 X_{ij}^{-1} = \operatorname{Tr}(S_2 (L + L\Delta L)_{(ij)}) \\
&\iff N_2 X_{ij}^{-1} = \operatorname{Tr}\left(((I \otimes S_2)(L + L\Delta L))_{(ij)}\right)
\end{aligned}$$

The last equivalence is simply the result of indices manipulation. Thus, we have

$$\nabla f_2(X) = -\nabla g_2(S_1) \iff X^{-1} = \frac{1}{N_2}\operatorname{Tr}_1((I \otimes S_2)(L + L\Delta L))$$

Similarly, by setting $J' = (\operatorname{vec}(S_1 \otimes E_{11}), \dots, \operatorname{vec}(S_1 \otimes E_{N_1 N_1}))$, we have that

$$\begin{aligned}
\nabla f_2(X)_{ij} = -(\nabla g_2(S_2))_{ij} &\iff N_1 X_{ij}^{-1} = (J'^\top \operatorname{vec}(-\nabla g(S)))_{ij} \\
&\iff N_1 X_{ij}^{-1} = \operatorname{vec}(S_1 \otimes E_{ij})^\top \operatorname{vec}(L + L\Delta L) \\
&\iff N_1 X_{ij}^{-1} = \operatorname{Tr}((S_1 \otimes E_{ij})(L + L\Delta L)) \\
&\iff N_1 X_{ij}^{-1} = \left(\sum\nolimits_{k,\ell=1}^{N_1} S_{1k\ell}(L + L\Delta L)_{(\ell k)}\right)_{ij} \\
&\iff N_1 X_{ij}^{-1} = \left(\sum\nolimits_{\ell=1}^{N_1} ((S_1 \otimes I)(L + L\Delta L))_{(\ell\ell)}\right)_{ij}
\end{aligned}$$

Hence,

$$\nabla f_{S_1}(X) = -\nabla g_{S_1}(S_2) \iff X^{-1} = \frac{1}{N_1}\operatorname{Tr}_2\left((S_1 \otimes I)(L + L\Delta L)\right),$$

which proves Prop. 3.1. □

B Efficient updates for KRK-PICARD

The updates to L_1 and L_2 are obtained efficiently through different methods; hence, the proof to Thm. 3.3 is split into two sections. We write

$$\Theta = \frac{1}{n}\sum_{i=1}^{n} U_i L_{Y_i}^{-1} U_i^\top \quad \text{(or } \Theta = U_i L_{Y_i}^{-1} U_i^\top \text{ for stochastic updates)}$$

so that $\Delta = \Theta - (I + L)^{-1}$. Recall that $(A \otimes B)_{(ij)} = a_{ij} B$.

B.1 Updating L_1

We wish to compute $X = \mathrm{Tr}_1\big((I \otimes L_2^{-1})(L\Delta L)\big)$ efficiently. We have

$$\begin{aligned}
X_{ij} &= \mathrm{Tr}\left[((I \otimes L_2^{-1})(L\Delta L))_{(ij)}\right] \\
&= \mathrm{Tr}\left[L_2^{-1}(L\Delta L)_{(ij)}\right] \\
&= \mathrm{Tr}\left[L_2^{-1}\sum\nolimits_{k,\ell=1}^{N_1} L_{(ik)}\Delta_{(k\ell)}L_{(\ell j)}\right] \\
&= \sum\nolimits_{k,\ell=1}^{N_1} L_{1ik}L_{1\ell j}\,\mathrm{Tr}(L_2^{-1}L_2\Delta_{(k\ell)}L_2) \\
&= \sum\nolimits_{k,\ell=1}^{N_1} L_{1ik}L_{1\ell j}\underbrace{\mathrm{Tr}(\Theta_{(k\ell)}L_2)}_{A_{k\ell}} - \underbrace{\mathrm{Tr}((I+L)^{-1}_{(k\ell)}L_2)}_{B_{k\ell}} \\
&= (L_1AL_1 - L_1BL_1)_{ij}.
\end{aligned}$$

The $N_1 \times N_1$ matrix A can be computed in $\mathcal{O}(n\kappa^3 + N_1^2N_2^2)$ time simply by pre-computing Θ in $\mathcal{O}(n\kappa^3)$ and then computing all N_1^2 traces in $\mathcal{O}(N_2^2)$ time. When doing stochastic updates for which Θ is sparse with only κ^2 non-zero coefficients, computing A can be done in $\mathcal{O}(N_1^2\kappa^2 + \kappa^3)$.

By diagonalizing $L_1 = P_1D_1P_1^\top$ and $L_2 = P_2D_2P_2^\top$, we have $(I+L)^{-1} = PDP^\top$ with $P = P_1 \otimes P_2$ and $D = (I + D_1 \otimes D_2)^{-1}$. P_1, P_2, D_1, D_2 and D can all be obtained in $\mathcal{O}(N_1^3 + N_2^3 + N_1N_2)$ as a consequence of Prop. 2.1. Then

$$\begin{aligned}
B_{ij} &= \mathrm{Tr}((I+L)^{-1}_{(ij)}L_2) \\
&= \sum_k \mathrm{Tr}(P_{(ik)}D_{(kk)}P_{(kj)}^\top L_2) \\
&= \sum_k P_{1ik}P_{1jk}\,\mathrm{Tr}(P_2D_{(kk)}P_2^\top P_2D_2P_2^\top) \\
&= \sum_k P_{1ik}P_{1jk}\underbrace{\mathrm{Tr}(D_{(kk)}D_2)}_{\alpha_k}.
\end{aligned}$$

Let $\widehat{D} = \mathrm{diag}(\alpha_1, \ldots, \alpha_{N_1})$, which can be computed in $\mathcal{O}(N_1N_2)$. Then $L_1BL_1 = P_1D_1\widehat{D}D_1P_1$ is computable in $\mathcal{O}(N_1^3 + N_2^3)$.

Overall, the update to L_1 can be computed in $\mathcal{O}(n\kappa^3 + N_1^2N_2^2 + N_1^3 + N_2^3)$ time, or in $\mathcal{O}(N_1^2\kappa^2 + \kappa^3 + N_1^3 + N_2^3)$ time if the updates are stochastic. Moreover, if Θ is sparse with only z non-zero coefficients (for stochastic updates $z = \kappa$), A can be computed in $\mathcal{O}(\kappa^2)$ space, leading to an overall $\mathcal{O}(z^2 + N_1^2 + N_2^2)$ memory cost.

B.2 Updating L_2

We wish to compute $X = \mathrm{Tr}_2\big[(L_1^{-1} \otimes I)(L\Delta L)\big]$ efficiently.

$$\begin{aligned}
X &= \sum\nolimits_{i=1}^{N_1}\big((L_1^{-1} \otimes I)(L\Delta L)\big)_{(ii)} \\
&= \sum\nolimits_{i=1}^{N_1}\big((I \otimes L_2)(\Theta - (I+L)^{-1})(L_1 \otimes L_2)\big)_{(ii)} \\
&= \sum\nolimits_{i,j=1}^{N_1} L_{1ij}L_2\Theta_{(ij)}L_2 - \sum_{i=1}^{N_1}((I \otimes L_2)(I+L)^{-1}(L_1 \otimes L_2))_{(ii)} \\
&= \underbrace{L_2\sum\nolimits_{i,j=1}^{N_1} L_{1ij}\Theta_{(ij)}L_2}_{A} - \underbrace{\sum\nolimits_{i=1}^{N_1}((I \otimes L_2)(I+L)^{-1}(L_1 \otimes L_2))_{(ii)}}_{B}
\end{aligned}$$

The matrix A can be computed in $\mathcal{O}(n\kappa^3 + N_1^2N_2^2 + N_2^3)$ time. As before, when doing stochastic updates A can be computed in $\mathcal{O}(N_1^2\kappa^2 + \kappa^3 + N_2^3)$ time and $\mathcal{O}(N_2^2 + N_1^2 + \kappa^2)$ space due to the sparsity of Θ.

Regarding B, as all matrices commute, we can write

$$(I \otimes L_2)(I + L)^{-1}(L_1 \otimes L_2) = (P_1 \otimes P_2)\Lambda(P_1 \otimes P_2)$$

where $\Lambda = (I \otimes D_2)(I + D_1 \otimes D_2)^{-1}(D_1 \otimes D_2)$ is diagonal and is obtained in $\mathcal{O}(N_1^3 + N_2^3 + N_1 N_2)$. Moreover,

$$B = \sum\nolimits_{i=1}^{N_1} (P\Lambda P^\top)_{(ii)} = P_2 \left(\sum\nolimits_{i,k=1}^{N_1} P_{1ik} \Lambda_{(kk)} P_{1ik} \right) P_2^\top,$$

which allows us to compute B in $\mathcal{O}(N_1^2 N_2 + N_2^3 + N_1^3)$ total.

Overall, we can obtain X in $\mathcal{O}(n\kappa^3 + N_1^2 N_2^2 + N_1^3 + N_2^3)$ or in $\mathcal{O}(N_1^2 \kappa^2 + N_1^2 N_2 + N_1^3 + N_2^3)$ for stochastic updates, in which case only $\mathcal{O}(N_1^2 + N_2^2 + \kappa^2)$ space is necessary.

C Proof of validity for joint updates

In order to minimize the number of matrix multiplications, we equivalently (due to the properties of the Frobenius norm) minimize the equation

$$\|L^{-1} + \Delta - X \otimes Y\|_F^2 \tag{11}$$

and set $\begin{cases} L_1' \leftarrow L_1 X L_1 \\ L_2' \leftarrow L_2 Y L_2. \end{cases}$.

Theorem C.1. *Let $L \succ 0$. Define $R := [\mathrm{vec}(L_{(11)})^\top; \ldots; \mathrm{vec}(L_{(N_1 N_1)})^\top]_{i,j=1}^{N_1} \in \mathbb{R}^{N_1 N_1 \times N_2 N_2}$.*

Suppose that R has an eigengap between its largest singular value and the next, and let u, v, σ be the first singular vectors and value of R. Let $U = \mathrm{mat}(u)$ and $V = \mathrm{mat}(v)$. Then U and V are either both positive definite or negative definite.

Moreover, for any value $\alpha \neq 0$, the pair $(\alpha U, \sigma/\alpha V)$ minimizes $\|L - X \otimes Y\|_F^2$.

The proof is a consequence of [22, Thm. 11]. This shows that if L is initially positive definite, setting the sign of α based on whether U and V are positive or negative definite[3], and updating

$$\begin{cases} L_1 \leftarrow \alpha\, L_1 U L_1 \\ L_2 \leftarrow \sigma/\alpha\, L_2 V L_2 \end{cases}$$

maintains positive definite iterates. Given that if $L_1 \succ 0$ and $L_2 \succ 0$, $L_1 \otimes L_2 \succ 0$, a simple induction then shows that by choosing an initial kernel estimate $L \succ 0$, subsequent values of L will remain positive definite.

By choosing α such that the new estimates L_1 and L_2 verify $\|L_1\| = \|L_2\|$, we verify all the conditions of Eq. 8.

C.1 Algorithm for joint updates

Theorem C.1 leads to a straightforward iteration for learning matrices L_1 and L_2 based on the decomposition of the Picard estimate as a Kronecker product.

Algorithm 3 JOINT-PICARD iteration

Input: Matrices L_1, L_2, training set T, step-size $a \geq 1$.
for $i = 1$ **to** maxIter **do**
  $U, \sigma, V \leftarrow$ power_method$(L^{-1} + \Delta)$ to obtain the first singular value and vectors of matrix R.
  $\alpha \leftarrow \mathrm{sgn}(U_{11}) \sqrt{\sigma \|L_2 V L_2\| / \|L_1 U L_1\|}$
  $L_1 \leftarrow L_1 + a(\alpha\, L_1 U L_1 - L_1)$
  $L_2 \leftarrow L_2 + a(\sigma/\alpha\, L_2 V L_2)$
end for
return (L_1, L_2)

[3] This can easily be done simply by checking the sign of the first diagonal coefficient of U, which will be positive if and only if $U \succ 0$.