set to 1 and halved when necessary, as per the algorithm described in (Gillenwater et al., 2014); we used the code of Gillenwater et al. (2014) for our EM implementation[7].

3.2. Synthetic tests

In each experiment, we sample n training sets from a base DPP of size N, then learn the DPP using EM and the Picard iteration. We initialize the learning process with a random positive definite matrix L_0 (or K_0 for EM) drawn from the same distribution as the true DPP kernel.

Specifically, we used two matrix distributions to draw the true kernel and the initial matrix values from:

- `BASIC`: We draw the coefficients of a matrix M from the uniform distribution over $[0, \sqrt{2}]$, then return $L = MM^\top$ conditioned on its positive definiteness.
- `WISHART`: We draw L from a Wishart distribution with N degrees of freedom and an identity covariance matrix, and rescale it with a factor $\frac{1}{N}$.

Figures 1, 2 and 3 show the log-likelihood as a function of time for different parameter values when both the true DPP kernel and the initial matrix L_0 were drawn from the `BASIC` distribution. Tables 1 and 2 show the final log-likelihood and the time necessary for each method to reach 99% of the optimal log likelihood for both distributions and parameters $n = 5000$, $a = 5$.

As shown in Figure 1, the difference in time necessary for both methods to reach a good approximation of the final likelihood (as defined by best final likelihood) grows drastically as the size N of the set of all elements $\{1, 2, \dots, N\}$ increases. Figure 2 illustrates the same phenomenon when N is kept constant and n increases.

Finally, the influence of the parameter a on convergence speed is illustrated in Figure 3[8]. Increasing a noticeably increases Picard's convergence speed, as long as the matrices remain positive definite during the Picard iteration.

The greatest strength of the Picard iteration lies in its initial rapid convergence: the log-likelihood increases significantly faster for the Picard iteration than for EM. Although for small datasets EM sometimes performs better, our algorithm provides substantially better results in shorter timeframes when dealing with larger datasets.

Overall, our algorithm converges to 99% of the optimal log-likelihood (defined as the maximum of the log-likelihoods returned by each algorithm) significantly faster than the EM algorithm for both distributions, particularly when dealing with large values of N.

Thus, the Picard iteration is preferable when dealing with large ground sets; it is also very well-suited to cases where larger amounts of training data are available.

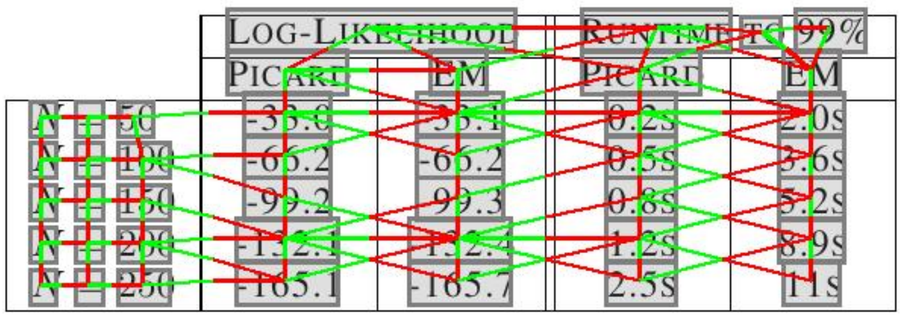
[7]These experiments were run with MATLAB, on a Linux Mint system, using 16GB of RAM and an i7-4710HQ CPU @ 2.50GHz.

[8]In the cases where $a > 1$, a safeguard was added to check that the matrices returned by our algorithm were positive definite.

Table 1. Final log-likelihoods and time necessary for an iteration to reach 99% of the optimal log likelihood for both algorithms when using `BASIC` distribution for true and initialization matrices (training set size of 5,000, $a = 5$).

	LOG-LIKELIHOOD		RUNTIME TO 99%	
	PICARD	EM	PICARD	EM
$N = 50$	-15.5	-15.5	17.3s	30.7s
$N = 100$	-24.4	-24.2	143s	75.5s
$N = 150$	-32.5	-32.5	40.7s	84.0s
$N = 200$	-40.8	-41.2	51.1s	1,730s
$N = 250$	-45.7	-46.0	99.1s	2,850s

Table 2. Final log-likelihoods and time necessary for an iteration to reach 99% of the optimal log likelihood for both algorithms when using `WISHART` distribution for true and initialization matrices (training set size of 5,000, $a = 5$).



In each case, 70% of the baby registries in the product category were used for training; 30% served as test. The results presented in Figures 4 and 5 are averaged over 5 learning trials, each with different initial matrices; the parameter a was set equal to 1.3 for all iterations.

Similarly to its behavior on synthetic datasets, the Picard iteration provides overall significantly shorter runtimes when dealing with large matrices and training sets. As shown in Table 3, the final log-likelihoods are very close (on the order 10^{-2} to 10^{-4}) to those attained by the EM algorithm.

Using a moments-matching initialization leaves Picard's runtimes overall unchanged (a notable exception being the 'gear' category). However, EM's runtime decreases drastically with this initialization, although it remains significantly longer than Picard's in most categories.

3.3. Baby registries experiment

We tested our implementation on all 13 product categories in the baby registry dataset, using two different initializations:

- the aforementioned Wishart distribution
- the data-dependent moment matching initialization (MM) described in (Gillenwater et al., 2014)