GPU-Project

- l - number of points
  k - no. of clusters
  d - dimensional space

Expectation:

i) $$\gamma_{ik} = \frac{\pi_k N(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{k} \pi_j N(x_i \mid \mu_j \Sigma_j)}$$

coalesced read in x_i and write in γ_i

- launch l kernels each computing $\gamma_{i0}, \gamma_{i1}, \ldots \gamma_{ik-1}$ using above formula

Maximization:

i) $N_k = \sum_{i=1}^{l} \gamma_{ik}$, $\pi_k = \frac{N_k}{l}$

- launch k kernels to compute both at one go

ii) $$\mu_k = \frac{1}{N_k} \sum_{i=1}^{l} \gamma_{ik} x_i$$

- step 1: $\underset{(k \times 1)}{\hat{\mu}} = \underset{(l \times k)}{R^T} \underset{(l \times d)}{X}$

GPU matrix multiplication

parallelization : $k \times d$

→ step 2 : launching k kernels to divide by N_k

iii) $$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{l} r_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

* Diagonal approximation

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{l} r_{ik} \sum_{j=1}^{d} (x_{ij} - \mu_{kj})^2$$

step 1 : compute v with $l \times k \times d^2$ threads

Coalesced write ← $v_{jxyi} = (P[i,x] - mu[j,x])^2 r[i,j]$
$(k \times d \times d \times l)$

step 2 : Sum along first dimension of v

Coalesced read ← gives $\hat{\Sigma}_{(k \times d \times d \times 1)}$ which on

adjustment gives Σ

$$\Sigma_{(k \times (d \times d))} = \frac{1}{N_k} \hat{\Sigma}_{(k \times d \times d \times 1)}$$

parallelization : $k \times d \times d$

Total 6 kernels = 5 (M step) + 1 (E step)