CUDA-level Performance with Python-level Productivity for Gaussian Mixture Model Applications

Henry Cook, Ekaterina Gonina, Shoab Kamil, Gerald Friedland, David Patterson, Armando Fox

Gaussian Mixture Models (GMM)
- Probabilistic model for clustering data
- Assumes the distribution of observations follows a mixture of multidimensional Gaussian distributions
- Each Gaussian in the mixture has a mean ($\mu$) and a variance ($\sigma^2$) parameter, as well as a weight ($\pi$)

Example applications:
- Speech Recognition – speaker classification, acoustic modeling for speech recognition
- Computer Vision – image segmentation, hand writing recognition
- Biology – flow cytometry
- Data mining – topics in documents

GM Training (EM algorithm)
- Given a set of observations/events: find the maximum likelihood estimates of the set of Gaussian Mixture parameters ($\mu, \sigma^2, \pi$) and classify observations
- Expectation Maximization (EM) Algorithm
  - E step: Compute probabilities of events given model parameters
  - M step: Compute model parameters given probabilities
- Specialization of the covariance matrix computation
- Platform parameters (GPU):
  - $N$ – number of events, ~10K-100K
  - $D$ – event dimension, ~10-40
  - $M$ – number of Gaussians (clusters), ~1-128
  - Matrix is symmetric – only compute the lower $D/2 \times D/2$ cells

Example Application Code
- Perform GMM training within an outer loop that decreases number of clusters
- Select best “fitting” GMM – number of clusters that best describes the event data
- Used in speaker diarization – unsupervised identification of speakers in an audio sample
- Compute the probability of observing an event given the trained model
- Used in speech recognition to compute the observation probability of an audio sample

Future Work
- More intelligent code variant selection mechanism, given platform and problem parameters
  - Pull from existing database of best-performing code variants
  - Use machine learning to predict the best performing code variant
  - Expand framework to other applications (computer vision, data mining) and architectures (OpenCL, RISC-V)
  - Performance improvement of the GMM framework for particular application common use cases to reduce overhead
  - Create more specializers for other patterns in speech recognition applications