Meta learning & machine learning

- **cross-validation** and **ensemble learning** crucial for deploying machine learning applications
- **no comprehensive meta learning layer** in large-scale ML systems
- e.g. **hyperparameter selection**: find good $k$ for $k$-NN spam classifier
- variety of methods (e.g. hold-out tests, k-fold cross-validation, resampling)
- two common steps:
  1. generate samples from input data
  2. train and evaluate machine learning models on the samples

Distributed sampling with replacement

- requires distributed generation of a multinomial random variate $r$ from the distribution of the number of occurrences of the $m$ rows of the input matrix in a sample of size $s$
- hard to parallelize: components of $r$ not independent
- but: multinomial is partitionable by conditioning on the totals of subsets of its components
  \[
  r_i = \sum_{j=1}^{m} r_{ij} = \sum_{j=1}^{m} \frac{n_j}{M_j} \text{ for } i = 1, \ldots, m
  \]
- combination with skip-ahead PRNGs leads to recursive algorithm for consistent generation of particular components of $r$ in parallel without inter-machine communication
  - by consistently generating only the partitions required to find a particular component

Distributed, single-pass algorithm for sample generation from blocked matrices

- leverages 'skip-ahead' pseudo-random-number-generators for embarrassingly parallel & consistent sampling without inter-machine communication
- use same PRNG and same seed on every machine
- use index of row to sample as position in to look in random sequence
- algorithm proceeds in three phases that execute in a single pass over the data in MapReduce-like systems
- embedded user-defined sampling function allows adaption to different sample generation techniques
- implementations for most common techniques in publication (e.g. hold-out tests, k-fold cross-validation, bagging)

Evaluation

- algorithm implemented in Hadoop, Spark, Hive
- extensive evaluation on large, synthetic matrices with different sampling techniques
- linear scalability with data size, number of machines, number of samples
- dynamic sample composition for k-fold cross-validation decreases runtime and intermediate data size
- algorithm outperforms matrix-based sample generation in Spark and join-based sample generation in Hive