**Motivation**

Moore’s law (single CPUs) comes to an end, but number of cores increases.
Growing amounts of data and need for accurate and confident predictions in critical applications.

Effective parallelisations achieve the same confidence and error bounds as the underlying base learning algorithm in much shorter time.

Efficient in the sense of Nick’s Class (\(L(C)\)):

- **polynomially many processing units.**
- **black-box parallelisation scheme** for a broad class of learning algorithms.

**Radon Point**

**Definition:**
A Radon partitioning of a set \(S \subseteq \mathcal{F}\) is a pair \((A, B)\) such that \(A \cup B = S\) and \(A \cap B = \emptyset\) but \((A) \cap (B) \neq \emptyset\), where \((\cdot)\) is the convex hull. The **Radon number** of a space \(\mathcal{F}\) is the smallest \(r \in \mathbb{N}\) such that for all \(S \subseteq \mathcal{F}\) with \(|S| \geq r\) there is a Radon partition. A **Radon point** of a set \(S\) with Radon partition \(A, B \subseteq S\) is any \(r \in (A) \cap (B)\). The Radon number of \(\mathbb{B}^d\) is \(r = d + 2\).

**Algorithm 1: Radon Machine**

- **Require:** learning algorithm \(\mathcal{L}\), dataset \(D \subseteq \mathbb{X} \times \mathbb{Y}\), Radon number \(r \in \mathbb{N}\), and parameter \(h \in \mathbb{N}\)

- **Ensure:** hypothesis \(f \in \mathcal{F}\)

  1. divide \(D\) into \(2^h\) iid subsets \(D_i\) of roughly equal size
  2. run \(\mathcal{L}\) in parallel to obtain \(f_i = \mathcal{L}(D_i)\)
  3. \(N = \{f_1, \ldots, f_{2^h}\}\)
  4. for \(i = h - 1, \ldots, 1\) do
  5. partition \(S\) into \(2^h\) iid subsets \(S_{i_1}, \ldots, S_{i_{2^h}}\) of size \(r\) each
  6. calculate \(c(S_{i_1}), \ldots, c(S_{i_{2^h}})\)
  7. \(c(S) = \{c(S_{i_1}), \ldots, c(S_{i_{2^h}})\}\) in parallel
  8. end for
  9. return \(c(S)\)

**Theoretical Analysis**

**Theorem:** given a consistent and efficient regularized risk minimization algorithm \(\mathcal{L}\) with hypothesis space \(\mathcal{F}\) with finite Radon number \(r\), and Radon number upper bounded by polylogarithmic function of \(N^2(\varepsilon, \Delta)\), then the Radon Machine \(R\) has polylogarithmic runtime on quasi-polynomially many processors.

**Empirical Evaluation**

**Theoretical Analysis**: 
- **Regularised risk minimisation algorithm** \(\mathcal{L}\) consistent: \((\varepsilon, \Delta)\)-guarantee.
- Efficient: \(N^2(\varepsilon, \Delta) = \varepsilon^2 N^2(\varepsilon, \Delta) = 2^h N^2(\varepsilon, \Delta)\).
- Runtime Complexity: \(T_z(M) = \mathcal{O}(M^e)\).
- Sample Complexity: \(M \geq N^2(\varepsilon, \Delta) = \varepsilon^2 M^2(\varepsilon, \Delta)\).

**Empirical Evaluation**: 
- **Speedup over Base Learning Algorithm**
- **Comparison with Spark and Averaging**