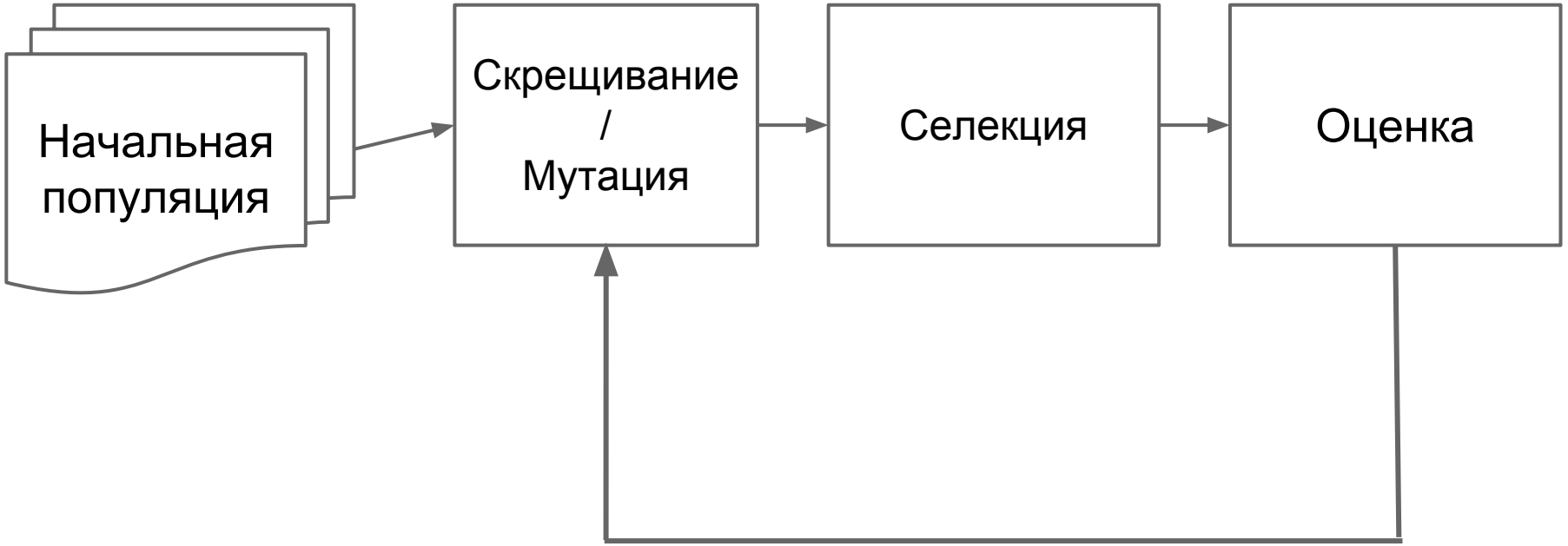
Развитие эволюционного программирования в Apache Spark

Толстопятов Всеволод, 344 группа
Научный руководитель:
Пахомов Е. А.,
Senior Software Engineer in Yandex/Alpine Data Labs,
Computer Science Center practices mentor

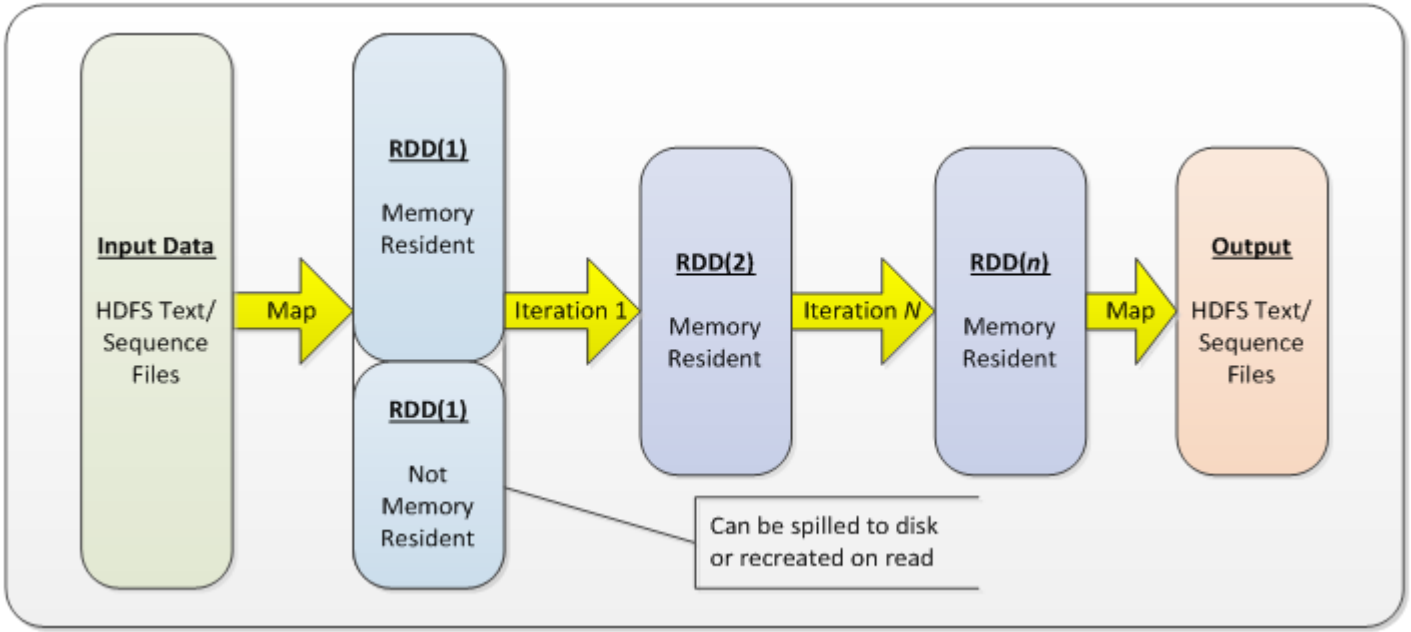
СПбГУ, Математико-механический факультет, 2015

Эволюционное программирование



Apache Spark

- Библиотека для распределенных вычислений и анализа данных под лицензией Apache 2.0.



Цель работы

- Исследовать типы задач, которые решаются с помощью эволюционного программирования
- Изучить существующие решения для эволюционных вычислений
- Реализовать основные примитивы ЭП
- Предоставить пользователям удобный интерфейс в Apache Spark для работы с ЭП
- Реализовать различные уровни параллелизма эволюционного процесса
- Отправить функционал в Spark как @Experimental* API

* — Может быть удалено или изменено в минорных версиях.

Фундаментальная теорема о шаблонах

Шаблон - подмножество множества всех возможных фенотипов с зафиксированными битами (частями).

Теорема утверждает, что шаблоны с приспособленностью выше средней распространяются экспоненциально.

Обоснование

Следствия фундаментальной теоремы о шаблонах:

- Избыточность (экспоненциальный рост подмножества решений)
- Линейное увеличение размера популяции экспоненциально ускоряет поиск решения
- Основная единица вычисление — шаблон, а не индивид
- Изменение процесса эволюции сильно изменяет множество допустимых шаблонов
- Основное ограничение — вычислительная мощность, которая компенсируется с помощью Spark

Классы задач

- Построение конечных автоматов
- Приближенное решение NP-полных задач

  В основном это SAT и HT
- Кластерный и регрессионный анализ
- Оптимизация функций

Ключевые свойства:

- Устойчивость к выбросам
- В контексте Spark — очень высокий уровень параллелизации (обычно это не так)

Инструменты и существующие решения

Инструменты

- Apache Spark
- Scala 2.10/2.11
- Tasmania — Spark/Hadoop кластер с 200+ вычислительных узлов, сделанный для YDF на основе CDH

Существующие решения

- Watchmaker (ex Mahout)
- ECJ

Реализация

- Необходимые инструменты для построения эволюционных алгоритмов: основные алгоритмы селекции, условия завершения, кроссоверы
- Простой API в сравнении с Watchmaker и ECJ
- Оптимальные параметры для задачи линейной регрессии и поиска минимума (методом роя частиц)

Линейная регрессия и рой частиц

Линейная регрессия:

- Дерево разбора
- При размерах популяции >=5000 Roulette Wheel Selection, иначе Stochastic Universal Sampling

Рой частиц:

- Чужие наработки
- Binary Tournament Selection

Реализация. Параллелизм

- По обучающей выборке
- По начальным поколениям (одна из самых эффективных моделей)
- По способам скрещивания и селекции (для проверки гипотез о процессе эволюции)
- Несколько версий островной модели
- Комбинация предыдущих

Результаты

- Изучен спектр задач, которые решаются эволюционным программированием
- Произведён обзор и сравнение существующих библиотек для эволюционного программирования
- Реализован основной функцонал для работы с эволюционным программированием в Apache Spark
- Реализованы два классических алгоритма машинного обучения
- Были исследованы и реализованы различные подходы к параллелизации процесса эволюции
- Начали переговоры с Cloudera для добавления результатов в Apache Spark