Введение в анализ данных: Заключение

Юля Киселёва
juliakiseleva@yandex-team.ru
Школа анализа данных

План на сегодня

- Алгоритм для кластеризации в неевклидовом пространстве
- Классификация (overview)
- Feature selection
- MapReduce (повторяем)

Residual sum of squares

- RSS – residual sum of squares

$$RSS_k = \sum_{x \in \omega_k} | \vec{x} - \mu(\omega_k) |^2$$

$$RSS = \sum_{k=1}^{K} RSS_k$$

$$\omega_k = cluster$$

$$\mu(\omega_k) = cetroid$$

Residual sum of squares(2)

- Минимизируем RSS

$$RSS_k(\vec{v}) = \sum_{\vec{x}\in\omega_k} |\vec{v} - \vec{x}|^2 = \sum_{x\in\omega_k}\sum_{m=1}^{M}(v_m - x_m)^2$$

$$\frac{\partial RSS_k(\vec{v})}{\partial v_m} = \sum_{x\in\omega_k} 2(v_m - x_m)$$

$$v_m = \frac{1}{|\omega_k|}\sum_{\vec{x}\in\omega_k} x_m$$

Нахождение оптимального числа кластеров

$$K = \arg\min_{k}[RSS_{\min}(K) + \lambda K]$$

Кластеризация в неевклидовом пространстве. GRGPF Algorithm

- **GRGPF** for its authors (V. Ganti, R. Ramakrishnan, J. Gehrke, A. Powell, and J. French)

- Организует кластер в иерархию в виде дерева
- Новый элемент может быть присоединен к кластеру к его нижним листьям дерева

Представление кластеров в GRGPF алгоритме

- Присоединение элемента в кластер => кластер увеличивается
- Большинство элементов кластера сохраняется на диск
- В основной памяти кластер представляется в виде вектора признаков (features)
- Для элемента p **из кластера:**

$$ROWSUM(\vec{p}) = \sum_{x \in \omega_k} | \vec{p} - \vec{x} |^2$$

Инициализация дерева

1. Кластеры организованы в виде дерева
2. Листья дерева могут быть очень большими (disk block или pages)
3. Инициализация: берется часть входных данных и кластеризуется иерархическим путем
4. Результат 3 = > дерево *T*
5. Данное дерево *T* не будет использоваться GRGPF
6. Полученное дерево балансируется
7. В одном кластере *N* элементов

GRGPF Алгоритм

- Данные из второго хранилища – ***p***
- ***P*** элемент добавляем в ближайший кластер
- Начинает с корня
- Доходим до листьев, в которых кластроид ближайший к ***p***
- Оцениваем: $ROWSUM(p) = ROWSUM(c) + Nd^2(p,c)$

Splitting clusters

- GRGPF накладывает ограничение на размер кластера:

$$\sqrt{ROWSUM(c)/N}$$

- Если радиус кластера превышает данную величину, то кластер делится на два:
  - Кластер загружается в основную память, разделяется на два путем минимизации ROWSUM
  - Признаки кластера вычисляются для обоих кластеров

Merging clusters

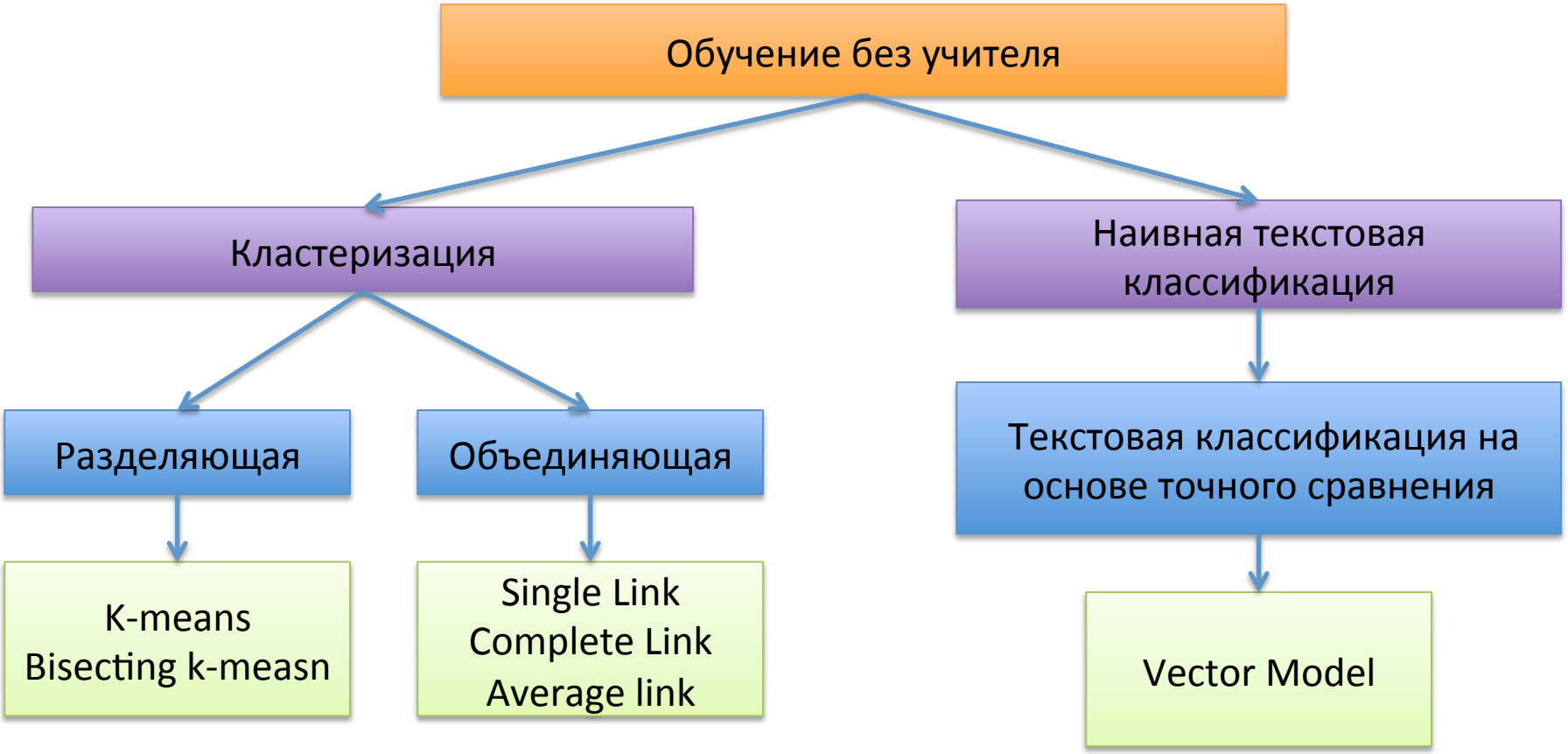
- Близкие кластеры объединяются (C1 и C2 в C)
- Кластройд для нового кластера C будет точка, которая максимально удалено от кластройда C1 или C2
- Например нужно посчитать ROWSUM для точки p из кластера C1 в новом кластере C:

$$ROWSUM_C(p) = ROWSUM_{c_1}(p) + N_{C_2}(d^2(p,c_1) + d^2(c_1,c_2)) + ROWSUM_{c_2}(c_2)$$

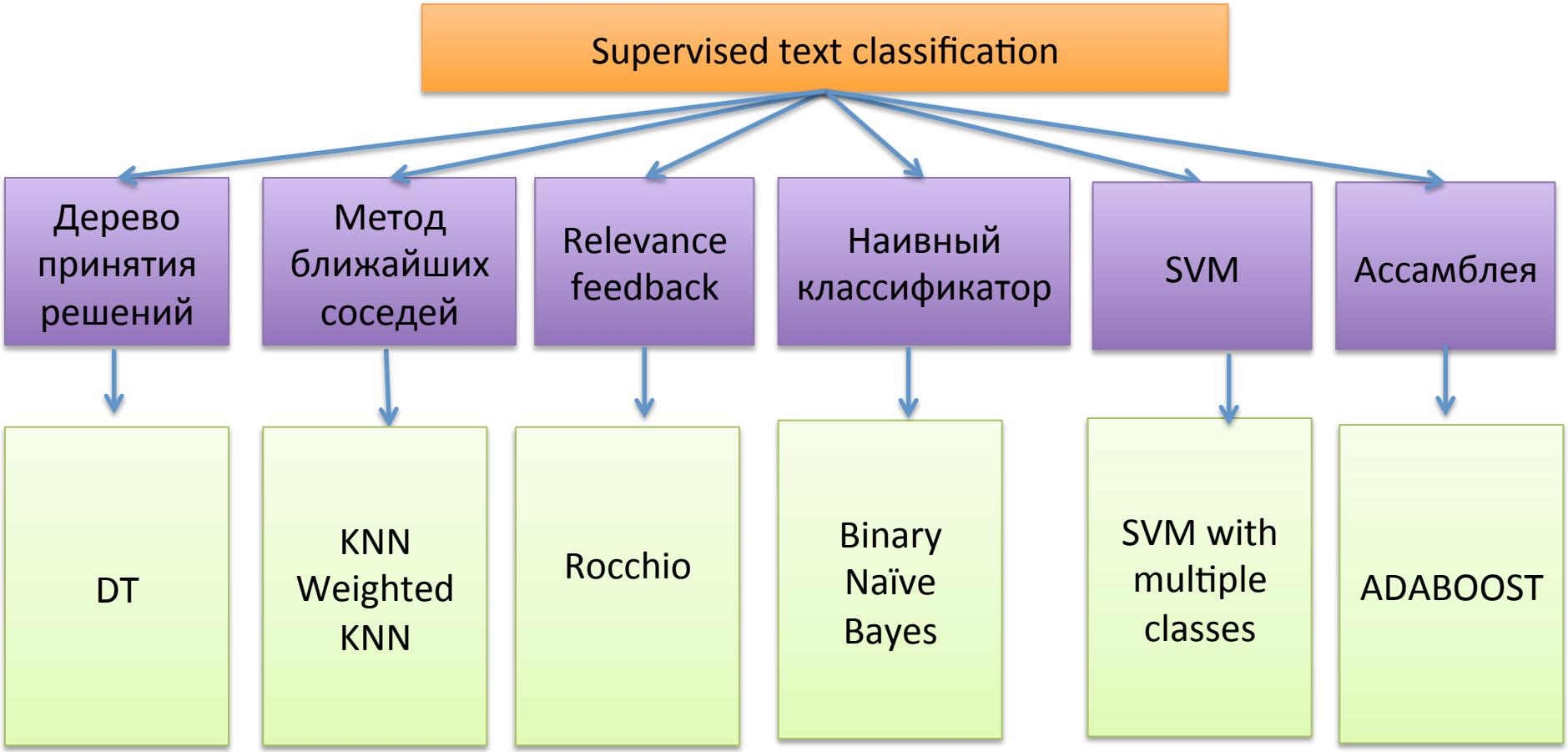
План на сегодня

- Алгоритм для кластеризации в неевклидовом пространстве
- Классификация (overview)
- Feature selection
- MapReduce (повторяем)

Классификация без учителя [Moderm IR. Ricardo Baeza-Yates]



Классификация с учителем [Moderm IR. Ricardo Baeza-Yates]



План на сегодня

- Алгоритм для кластеризации в неевклидовом пространстве
- Классификация (overview)
- Feature selection
- MapReduce (повторяем)

Оценивание признаков [2003 Feature Selection Challenge]

Метод оценки: Balance Error Rate (BER)

$$\mathrm{BER} \equiv \frac{1}{2}\left(\frac{\#\text{ positive instances predicted wrong}}{\#\text{ positive instances}} + \frac{\#\text{ negative instances predicted wrong}}{\#\text{ negative instances}}\right).$$

Оценка признаков. F-score (2)

$x_k, k = 1, \ldots, m,$ Входящий вектор признаков

n_+ and n_-, Число положительных и отрицательных Элементов в обучающей выборке

$$F(i) \equiv \frac{\left(\bar{x}_i^{(+)} - \bar{x}_i\right)^2 + \left(\bar{x}_i^{(-)} - \bar{x}_i\right)^2}{\frac{1}{n_+ - 1} \sum_{k=1}^{n_+} \left(x_{k,i}^{(+)} - \bar{x}_i^{(+)}\right)^2 + \frac{1}{n_- - 1} \sum_{k=1}^{n_-} \left(x_{k,i}^{(-)} - \bar{x}_i^{(-)}\right)^2},$$

$\bar{x}_i, \bar{x}_i^{(+)}, \bar{x}_i^{(-)}$ Это среднее по i-ому признаку по всей выборке, по положительным и отрицательным элементам

Методы выбора признаков на основе F-score

1. Считаем F-score для каждого признака
2. Выбираем несколько порогов для F-score (решение человека)
3. Для каждого порога:
   1. Удаляем признаки с низким F-score
   2. Случайным образом, разделяем на *X_training* and *X_valid*
   3. Обучаем классификатор на основе *X_training*
   4. Повторяем шаги 5 раз и выбираем порог с наименьшей ошибкой
4. Удаляем из обучающей выборки невалидные признаки

F-score and Random Forest

- Random Forest (RF) – метод классификации, возвращает важность каждого признака
- Основная идея – используется несколько (большое количество) деревьев принятия решений, каждое из которых строится на основе обучающей выборки. Вектор признаков выбирается случайно
- Предсказание определяется голосование, признается мнение большинства

F-score and Random Forest. Определение важности признака

- Разделяем обучающую выборку на две части (TS1 и TS2)
- Обучаемся на TS1
- Тестируем на TS2 = > Accuracy1
- Случайным образом переставляем значения для *l* – ого признака => accuracy2
- Важность признака = |Accuracy1 – Accuracy2|

F-score and Random Forest. Определение важности признака (2)

1. F-score (рассматривает только признаки с высоким F-score, как получено ранее)
2. Стартуем RF, на основе признаков из 1. RF возвращает ранк для каждого признака.
3. Убираем из обучающей выборки менее важные признаки.
4. Останавливаемся, если число признаков маленькое

Результаты. [2003 Feature Selection Challenge]

Dataset	ARCENE	DEXTER	DOROTHEA	GISETTE	MADELON
SVM	13.31	11.67	33.98	2.10	40.17
F+SVM	**21.43**	**8.00**	21.38	**1.80**	13.00
F+RF+SVM	21.43	8.00	**12.51**	1.80	13.00

План на сегодня

- Алгоритм для кластеризации в неевклидовом пространстве
- Классификация (overview)
- Feature selection
- MapReduce (повторяем)

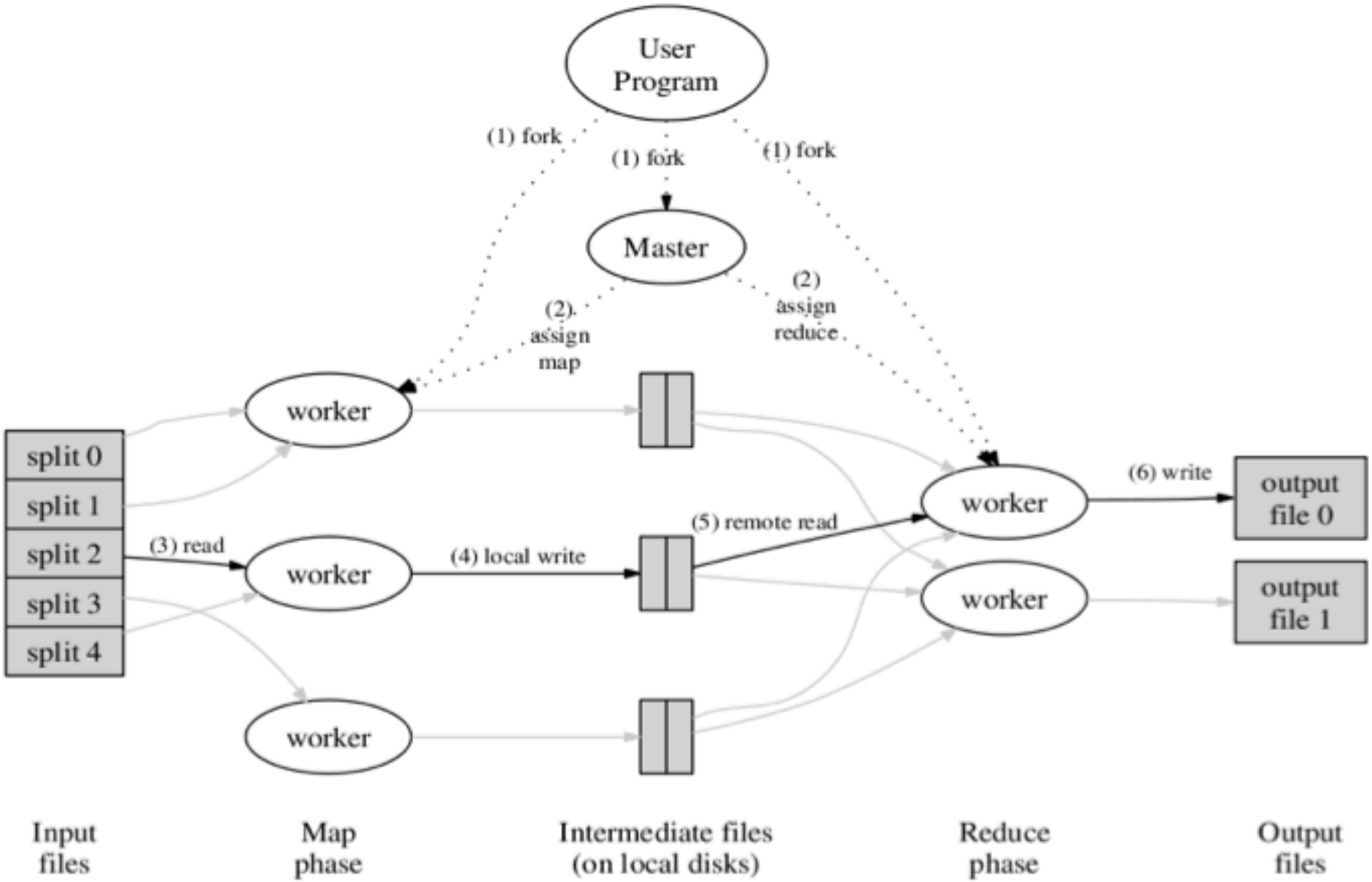
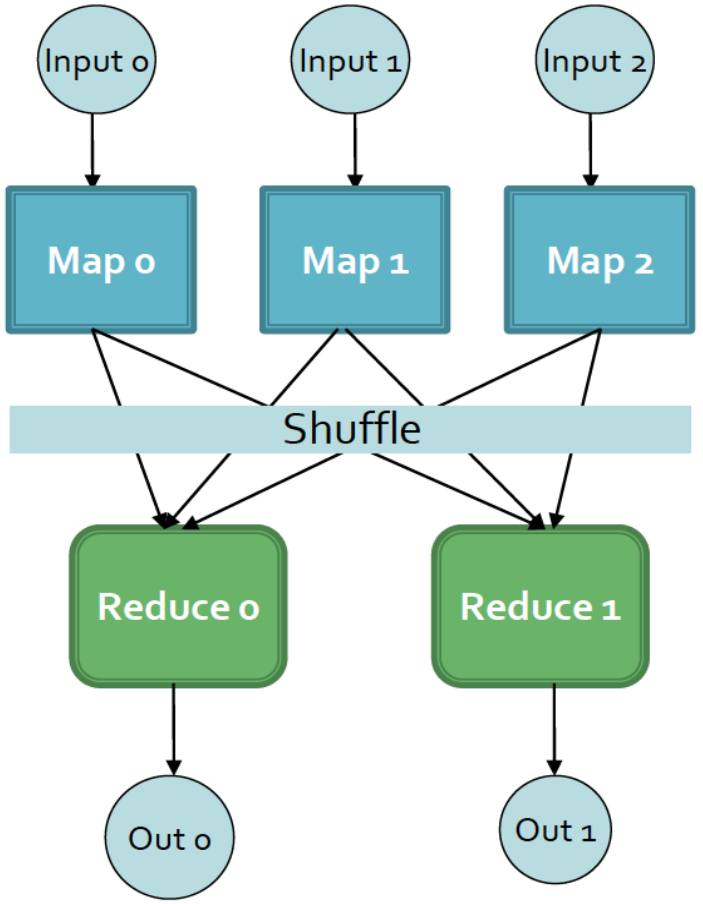


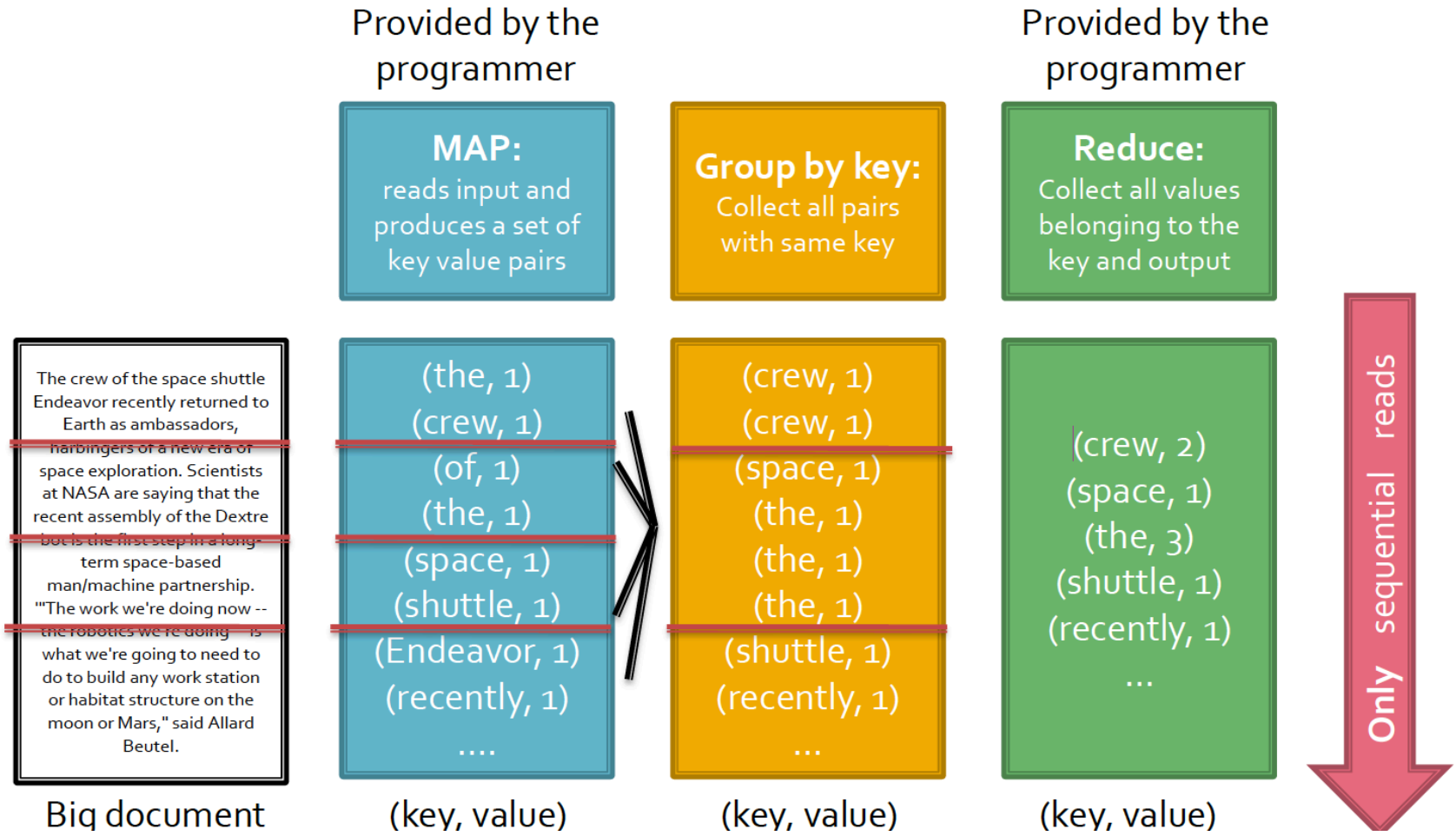
Figure 1: Execution overview

MapReduce: представление

- Читает данные
- Функция – **Map <k,v”>:**
  - Извлекает то, что Вам нужно
- Смешивает и сортирует
- Функция – **Reduce <k, v”> -> <k, v””>:**
  - Агрегирует, суммирует, фильтрует и трансформирует
- Пишет результат



Пример: подсчет статистики по словам



Резюме. Этапы эксперимента

1. **Формулировка задачи** (рассмотрели)
2. **Выбор алгоритма для анализа и методов оценки** и методов оценки (рассмотрели)
3. **Выбор обучающего и тестового множества** (рассмотрели)
4. **Feature selection (Выбор признаков)** (рассмотрели)
5. **Оценка полученных результатов** (рассмотрели)
6. **Вывод** (рассмотрели)