Trabajo Práctico 3: Map-Reduce

Martín Arjovsky 683/12
Ezequiel Darío Gambaccini 715/13
Silvio Vileriño 106/12

Julio 2014

1 Encontrar las mejores películas

Se requería obtener la lista de las 12 películas mejor rankeadas con más de 20 reviews. Para esto, lo que decidimos hacer fue que en el map se emitiera el id del producto con un objeto que contenía el puntaje de la review y una variable count inicializada a 1.
En el siguiente paso, reduce, lo que se hace es sumar todos los objetos puntaje, cantidad y emitir el resultado.
Finalmente, en el finalize, se filtra aquellas peliculas que tienen count ¿ 19. Para las películas que tienen más de 19 reseñas se devuelve la sumatoria de puntajes dividida la cantidad de reseñas.
Para las que tienen menos de 20 reseñas se devuelve -1.
Luego, usando un script de python para parsear la salida de runner.py, se filtra aquellas peliculas que su puntaje promedio es distinto de -1, y luego se las ordena en orden decreciente por puntaje, quedandose con las primeras 12.

Las películas que obtuvimos con sus respectivos puntajes y nombres (conseguidos a través del buscador de Amazon) fueron:

- B00000JSJ5, 5.0, All Creatures Great and Small, Series 2: Volumes 1-6
- B00004YKS6, 5.0, Genghis Blues
- B00004YKS7, 5.0, Genghis Blues
- B0002NY7UY, 5.0, Live in Concert (Dion)
- B0007GAEXK, 5.0, The Mole - The Complete First Season
- B0007Z4HAC, 5.0, Salsa Crazy Presents: Learn to Salsa Dance, Intermediate Series, Volume 1
- B000AOEPU2, 5.0, WWE: Bret "Hitman" Hart - The Best There Is, The Best There Was, The Best There Ever Will Be
- B000MCIADA, 5.0, A Reiki 1st, Aura and Chakra Attunement Performed

- B003YBGJ4S, 5.0, WELL worked out with Tannis
- B004LK24BI, 5.0, 50/50 Cardio and Weights with Angie Gorr
- B006JN87UC, 5.0, Transformers: Prime - Season One
- B008COIZHQ, 5.0, Genghis Blues
- B005FY0FPG, 4.987012987012987, Dream With Me in Concert
- B000M7XRC4, 4.977777777777778, The Venture Bros. - Season Two

A continuación se muestra el código de map, reduce y finalize respectivamente.

```
function () {
  data = {};
  data.score = parseFloat(this.score,10);
  data.count = 1;
  emit(this.productId, data);
}

function (key, values) {
  var res = {};
  res.score = 0;
  res.count = 0;
  for (var i = values.length - 1; i >= 0; i--) {
    var data = values[i];
    res.score += data.score;
    res.count += data.count;
  };
  return res;
}

function (key, reducedVal) {
  if (reducedVal.count > 19) {
    return reducedVal.score/reducedVal.count;
  } else{
    return -1;
  };
}
```

2 Palabras más frecuentes

Para obtener las 5 palabras más usadas de cada puntaje, en la función de map se realizan varias operaciones:

1. Se filtran los carácteres del texto de la review para que queden solo los alfanumericos, y se los pasa a minuscula.
2. Se divide el texto en un array de palabras y luego se aplica una función para filtrar aquellas palabras que están en el array de stop words.

3. Finalmente, se cuentan las apariciones de cada palabra en un objeto, y se emite el valor del puntaje y el objeto que cuenta ocurrencias.

En el reduce, simplemente se genera un nuevo objeto con las ocurrencias de cada palabra de ese puntaje particular.
En el finalize, se invierten los valores del objeto, es decir, se devuelve un objeto que contiene numeros como claves y listas de palabras como valor. Los numeros representan la cantidad de apariciones de las palabras de la lista.
Finalmente, las palabras más usadas para las reviews, divididas por puntaje, ocurrencias, y palabras, es:
A continuación se muestra el código de map, reduce y finalize respectivamente.

```
1  function () {
2
3    var data = this.text.slice(0);
4
5    data = data.replace(/\W/g, ' ').toLowerCase(); //remueve
         caracteres no alfanumericos
6
7    var stop_words = ["","a","able","about","across","after","all","
         almost","also","am","among","an",
8    "and","any","are","as","at","be","because","been","but","by","can
         ","cannot",
9    "could","dear","did","do","does","either","else","ever","every","
         for","from",
10   "get","got","had","has","have","he","her","hers","him","his","how
         ","however",
11   "i","if","in","into","is","it","its","just","least","let","like",
         "likely","may",
12   "me","might","most","must","my","neither","no","nor","not","of","
         off","often",
13   "on","only","or","other","our","own","rather","said","say","says"
         ,"she","should",
14   "since","so","some","than","that","the","their","them","then","
         there","these",
15   "they","this","tis","to","too","twas","us","wants","was","we","
         were","what",
16   "when","where","which","while","who","whom","why","will","with","
         would","yet","you","your"];
17
18   var clear_stopwords = function (elem) {
19     return (stop_words.indexOf(elem) === -1);
20   }
21
22   var filtered = data.split(" ").filter(clear_stopwords);
23
24   var dict = new Object();
25
26   for (var i = filtered.length - 1; i >= 0; i--) {
27     var s = filtered[i];
28     if( s in dict)  {
29       dict[s] += 1;
30     }
31     else {
32       dict[s] = 1;
33     }
```

```
  };

  emit(this.score, tojson(dict));
}

function (key, values) {

  var res = new Object();

  for (var i = values.length - 1; i >= 0; i--) {
    var data = JSON.parse(values[i]);
    for (var key in data){
      if (data.hasOwnProperty(key)){
        if( key in res)  {
          res[key] += 1;
        }
        else {
          res[key] = 1;
        }
      }
    }
  };

  return tojson(res);
}

function (key, reducedVal) {
  reducedVal = JSON.parse(reducedVal);
  var top_words = new Object();

  for(var word in reducedVal){
    if (reducedVal.hasOwnProperty(word)) {
      var count = reducedVal[word];
      if (count in top_words){
        top_words[count].push(word);
      }
      else{
        top_words[count] = [word];
      }
    }
  }
  return tojson(top_words);
}
```

3 Helpfulness y Longitud

Este fue el ejercicio más sencillo de los tres.
En el map, simplemente se emite el helpfulness y la longitud de la review.
Luego, en el reduce, se calcula la longitud promedio de las reviews, sumando sus longitudes y dividiendolas por la cantidad de valores en ese reduce.
A continuación se muestra el código de map y reduce respectivamente.

```
function () {
  emit(this.helpfulness, this.text.length);
}
```

```
4
5  function (key, values) {
6      avg_length = 0;
7      for (var i=0; i < values.length; i++){
8          avg_length += values[i];
9      }
10     return avg_length/values.length;
11 }
```

La data luego fue parseada para realizar el análisis del punto 4. En este caso se midió la helpfulness como el cociente entre los votos de que una review era helpful y la cantidad total de votos. Se evaluó el coeficiente de correlación lineal de Pearson entre la helpfulness y la longitud promedio de las reviews, dando un valor de 0.3159. Esto indica que las variables están positivamente correlacionadas, indicando que es probable que cuando una crezca, la otra lo haga también.
Sin embargo es importante mostrar que no parece haber una correlación lineal entre estos factores, como se evidencia en el siguiente gráfico, por esto es que concluimos que otras métricas pueden ser más útiles para medir la influencia de un dato sobre el otro. De todas maneras este coeficiente nos da la indicación (soportada por el gráfico) de que estas variables no son independientes, y de que se puede extraer información de una a partir de la otra.