MapReduceを活用したコンテンツベースレコメンデーションのための分散処理システム

山本 努† 矢吹 太朗 佐久田 博司
青山学院大学 理工学部 情報テクノロジー学科‡

1 序論

教育機関にインターネットに接続された端末の導入が進み，従来の教師と学習者の対面授業に，e-learningの利用を組み込んだBlended learningが増加している．Blended learningには次のようなメリットがある．

1. 対面授業のメリットであるドロップアウトしにくいという点と，e-learningのメリットであるフィードバックが確認できる点や個々の能力に合わせた学習方法が選択できるという点を活かしている．
2. 学習者はインターネットに接続された端末を利用することが可能なため，授業中に発生した疑問について教師の個別指導に頼らずWebからの情報収集によって解決することができる．

しかし，一方で次のような問題があり，その解決が求められている．

1. 教師は学習者のWeb利用状況を把握できない．
2. Web上の情報は学習を考慮して整理されているわけではないため，学習者が最適な情報を探すことが困難である．

1に関してはBlended learngにおいて学習者のWeb履歴やWebページのリンク先を取得して調査し，支援をしていく研究が行われている[5]．2に関してはWebページの推薦に関する研究が挙げられる．特にコンテンツベースの手法を用いる研究は，Webページの内容を考慮可能なため，学習者に対し適切な情報を提示できる可能性がある．だが，これらの研究では，各文書の言語処理や文書間の類似度計算などために計算負荷が大きくなることが懸念される．矢島らは，類似度計算において結果の妥当性の範囲内で，計算量を減らすことでの高速化を検討している[6]．しかし，Blended learuningでは最も適切な情報の提示が必要であり，計算負荷を下げるために精度を落とすことはできるかぎり回避したい．

そこで本研究では，Webを利用するBlended learningにおける，Webページの推薦システムを構築する基盤として，大量の文書を分散処理するシステムの構築を目的とする．対象とするタスクは，文書内容に基づく推薦（コンテンツベース・レコメンデーション）のために必要な，文書間の類似度計算とする．

2 手法

2.1 分散処理

Google Inc.によって2004年に提唱された分散処理技術であるMapReduce [1]のオープンソース実装Hadoop [2, 3]を採用する．MapReduceは，処理を分散するMap処理と，分散した処理を集約するReduce処理で構成されている．Map処理はキーと値の組み合わせを受け取り、新たなキーと値を生成する．Reduce処理はMap処理が出力した結果の値を，キーごとにグループ化し，一つの値を出力する．

2.2 類似度計算

類似度計算の手法としてtf-idf法とベクトル空間モデルを採用する．この手法は文書の類似度を算出する際に幅広く用いられ，類似レポート検出などにも利用されている[4]．tf-idf法は単語の出現頻度と逆出現頻度を用いて文書の特徴を表している単語を抽出するアルゴリズムである．ベクトル空間モデルとは文書を多次元のベクトルで表現し，ベクトルのなす角度を類似度の指標として見る手法である．tf-idf法で求めた値を要素とした多次元ベクトルを比較することにより，類似度を求めることができる．

2.3 MapReduceの手順

与えられた文書集合内の，すべての文書の組みの類似度を，MapReduceで計算する方法を示す．以下では，キーと値の組みを「(キー, 値)」, (キー, 値) のリストを「list(キー, 値)」と表記する．「tf-idf'」は，文書内で規格化したtf-idfである．つまり，ある文書内に出現するすべての単語の「tf-idf'」を並べたベクトルのノルムは1である．

Distributed computing system with MapReduce for content-based recommendation.

† Tsutomu YAMAMOTO (a5806082@aoyama.jp)
‡ Department of Integrated Information and Technology, College of Science and Engineering, Aoyama Gakuin University

Copyright © 2010 Information Processing Society of Japan.
All Rights Reserved.

1. 入力: (文書 ID, コンテンツ)
2. Map: ⟶ (単語, (文書 ID, 1))
   入力された文書を単語に分ける．あとで集計するために，数 1 を合わせて出力する
3. Reduce: ⟶ (単語, list(文書 ID, tf))
   文書 ID と tf（文書内の単語頻度）のリストを単語ごとにまとめる
4. Map: ⟶ (文書 ID, (単語, tf-idf))
   リストの長さが df であることを利用して，tf-idf を計算し，文書 ID をキーにして出力する
5. Reduce: ⟶ (文書 ID, list(単語, tf-idf))
   単語と tf-idf の組みのリストを文書 ID ごとにまとめる
6. Map: ⟶ (文書 ID, list(単語, tf-idf')),
   ⟶ (単語, (文書 ID, tf-idf'))
   tf-idf を規格化し，単語をキーにして出力する
7. Reduce: ⟶ (単語, list(文書 ID, tf-idf'))
   文書 ID と規格化した tf-idf 値のリストを単語ごとにまとめる
8. Map: ⟶ (文書 ID-a, 文書 ID-b, list(tf-idf'-a tf-idf'-b))
   リストを 2 重のループで処理し，規格化された tf-idf の積を出力する
9. Reduce: ⟶ ((文書 ID-a, 文書 ID-b), 類似度)
   規格化された tf-idf の組みの積の合計が類似度となる

2.4 具体例

下記の例文を用いた処理の様子を表 1 と表 2 に示す．

A Time is money.
B Money comes and goes.
C Money goes, love goes.

表 1 手順 1, 2 が終了した時点での出力

Map 処理		Reduce 処理	
キー	値	キー	値
time	(A, 1)	time	(A, 1)
is	(A, 1)	is	(A, 1)
money	(A, 1)	money	(A, 1)(B, 1)(C, 1)
money	(B, 1)	comes	(B, 1)
comes	(B, 1)	and	(B, 1)
and	(B, 1)	goes	(B, 1)(C, 2)
goes	(B, 1)	love	(C, 1)
money	(C, 1)		
goes	(C, 1)		
love	(C, 1)		
goes	(C, 1)		

表 2 手順 9 が終了した時点での出力

キー	値
(A, B)	0
(A, C)	0
(B, C)	0.149962

このように，MapReduce によって，文書集合内のすべての文書間の類似度が計算できる．

3 結論・課題

本研究では，文書集合内のすべての文書間の類似度を，分散処理技術である MapReduce を用いて計算する方法を示した．今後の解題として以下のようなことが挙げられる．

- スケーラビリティの調査
- オンラインで利用する方法の開発

参考文献

[1] Jeffrey Dean and Sanjay Ghemawa. MapReduce: Simplified data processing on large clusters. *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, pp. 137–150, 2004.

[2] Apache Software Foundation. Hadoop. `http://hadoop.apache.org/`.

[3] Tom White. *Hadoop: The Definitive Guide*. O'Reilly, 2009.

[4] 岩堀祐之, 舟橋健司, 伊藤宏隆, 石井直宏. 情報メディア教育における類似レポート判定システムの構築. 平成 14 年度情報処理教育研究集会講演論文集, pp. 654–657, 2002.

[5] 渡邉貴志, 矢吹太朗, 佐久田博司. Web 利用履歴のリアルタイムモニタリングによるクラスの学習状況把握ツールの開発. 信学技報 (電子情報通信学会技術研究報告 [教育工学]), pp. 37–42, 2008.

[6] 矢島健太郎, 井上潮. ソーシャルブックマークにおける文書解析を利用した類似文書および類似ユーザの推薦方法の提案. 電子情報通信学会第 18 回データ工学ワークショップ論文集, 2007, 2007.

Copyright © 2010 Information Processing Society of Japan.
All Rights Reserved.