

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5230421号  
(P5230421)

(45) 発行日 平成25年7月10日(2013.7.10)

(24) 登録日 平成25年3月29日(2013.3.29)

(51) Int.Cl. F I  
**G06F 17/30 (2006.01)**  
 G06F 17/30 210D  
 G06F 17/30 340A  
 G06F 17/30 412

請求項の数 11 (全 21 頁)

(21) 出願番号	特願2008-527069 (P2008-527069)	(73) 特許権者	507103802
(86) (22) 出願日	平成18年8月15日 (2006.8.15)		グーグル・インコーポレーテッド
(65) 公表番号	特表2009-505290 (P2009-505290A)		アメリカ合衆国・カリフォルニア・940
(43) 公表日	平成21年2月5日 (2009.2.5)		43・マウンテン・ビュー・アンフィシア
(86) 国際出願番号	PCT/US2006/031868		ター・パークウェイ・1600
(87) 国際公開番号	W02007/022199	(74) 代理人	100108453
(87) 国際公開日	平成19年2月22日 (2007.2.22)		弁理士 村山 靖彦
審査請求日	平成21年8月14日 (2009.8.14)	(74) 代理人	100064908
(31) 優先権主張番号	11/204,922		弁理士 志賀 正武
(32) 優先日	平成17年8月15日 (2005.8.15)	(74) 代理人	100089037
(33) 優先権主張国	米国 (US)		弁理士 渡邊 隆
前置審査		(74) 代理人	100110364
			弁理士 実広 信哉

最終頁に続く

(54) 【発明の名称】 集合の類似性に基づく拡張性に富むユーザクラスタリング

(57) 【特許請求の範囲】

【請求項1】

情報担体上に符号化されたコンピュータプログラムであって、

データ処理装置に、複数ユーザの各々についてそれぞれの興味集合を取得させることができる命令を有して、各興味集合はそれぞれのユーザがデータ処理システムとの相互作用を通して興味を示した品目を表わし、

更に、データ処理装置に、複数ユーザの各々について、それぞれの興味集合のk個のハッシュ値を決定させることができる命令を有して、第iのハッシュ値は、それぞれ他と異なるk個のハッシュ関数のうち対応する第iのハッシュ関数の下で、品目の母集団の上のk個の順列のうち第iの順列の下でのそれぞれの興味集合中の最小値であり、iは1とkの間の整数であり、kは1より大きい整数であり、

更に、データ処理装置に、複数ユーザの各々を、それぞれのユーザについて第iのクラスタは第iのハッシュ値によって表わされるように確立されたそれぞれのk個のクラスタの各々に割り当てさせることができる命令を有して、複数ユーザの各々のk個のクラスタへの割り当ては他のどのユーザについてのk個のクラスタへの割り当てにも無関係に行われる

ことを特徴とするコンピュータプログラム。

【請求項2】

データ処理装置に、ユーザの興味を表す行動をログに記録させることができる命令と、データ処理装置に、前記複数ユーザについて前記興味集合を生成するために前記ログを

使用させることができる命令と

をさらに含むことを特徴とする請求項 1 に記載のコンピュータプログラム。

【請求項 3】

データ処理装置に、前記複数ユーザの第 1 のユーザについて変化のあった興味集合を取得させることができる命令と、

データ処理装置に、変化のあった興味集合を使用する第 1 のユーザについて k 個のハッシュ値を決定させることができる命令と、

データ処理装置に、第 1 のユーザを、変化のあった興味集合を使用して決定される k 個のハッシュ値によって表わされるそれぞれの k 個のクラスタの各々にのみ、他の複数ユーザのクラスタへの割り当てを変化させることなく、割り当てさせることができる命令と

10

をさらに含むことを特徴とする請求項 1 に記載のコンピュータプログラム。

【請求項 4】

情報担体上に符号化されたコンピュータプログラムであって、

データ処理装置に、ユーザについて興味集合を取得させることができる命令を有していて、前記興味集合は前記ユーザがデータ処理システムとの相互作用を通して興味を示した品目を表わし、

更に、データ処理装置に、前記興味集合の k 個のハッシュ値を決定させることができる命令を有していて、第 i のハッシュ値は、それぞれ他と異なる k 個のハッシュ関数のうち対応する第 i のハッシュ関数の下で、品目の母集団の上の k 個の順列のうち第 i の順列の下での前記興味集合中の最小値であり、i は 1 と k の間の整数であり、k は 1 より大きい

20

整数であり、  
更に、データ処理装置に、前記ユーザを k 個のクラスタの各々に割り当てさせることができる命令を有していて、第 i のクラスタは第 i のハッシュ値により表わされることを特徴とするコンピュータプログラム。

【請求項 5】

前記興味集合は m 個の要素を持ち、

第 i のハッシュ値は一方向ハッシュ関数を m 回適用したときの最小値であり、各適用は第 i シード値と前記興味集合の m 個の要素のそれぞれ 1 つをハッシュする

ことを特徴とする請求項 4 に記載のコンピュータプログラム。

【請求項 6】

30

データ処理装置に、前記ユーザについての協調フィルタリングを実行するために、前記 k 個のクラスタを使用させることができる命令をさらに含む

ことを特徴とする請求項 4 に記載のコンピュータプログラム。

【請求項 7】

複数ユーザの各々についてそれぞれの興味集合を取得するステップを有していて、各興味集合はそれぞれのユーザがデータ処理システムとの相互作用を通して興味を示した品目を表わし、

更に、複数ユーザの各々について、それぞれの興味集合の k 個のハッシュ値を決定するステップを有していて、第 i のハッシュ値は、それぞれ他と異なる k 個のハッシュ関数のうち対応する第 i のハッシュ関数の下で、品目の母集団の上の k 個の順列のうち第 i の順列の下でのそれぞれの興味集合中の最小値であり、i は 1 と k の間の整数であり、k は 1 より大きい整数であり、

40

更に、複数ユーザの各々を、それぞれのユーザについて第 i のクラスタは第 i のハッシュ値によって表わされるように確立されたそれぞれの k 個のクラスタの各々に割り当てるステップを有していて、複数ユーザの各々の k 個のクラスタへの割り当ては他のどのユーザについての k 個のクラスタへの割り当てにも無関係に行われる

ことを特徴とするデータ処理システムの作動方法。

【請求項 8】

ユーザの興味を表す行動をログに記録するステップと、

前記複数ユーザについて前記興味集合を生成するために前記ログを使用するステップと

50

をさらに有していることを特徴とする請求項 7 に記載の方法。

【請求項 9】

ユーザについて興味集合を取得するステップを有していて、前記興味集合は前記ユーザがデータ処理システムとの相互作用を通して興味を示した品目を表わし、

更に、前記興味集合の  $k$  個のハッシュ値を決定するステップを有していて、第  $i$  のハッシュ値は、それぞれ他と異なる  $k$  個のハッシュ関数のうち対応する第  $i$  のハッシュ関数の下で、品目の母集団の上の  $k$  個の順列のうち第  $i$  の順列の下での前記興味集合中の最小値であり、 $i$  は 1 と  $k$  の間の整数であり、 $k$  は 1 より大きい整数であり、

更に、前記ユーザを  $k$  個のクラスタの各々に割り当てるステップを有していて、第  $i$  のクラスタは第  $i$  のハッシュ値により表わされる

ことを特徴とするデータ処理システムの作動方法。

10

【請求項 10】

前記興味集合は  $m$  個の要素を持ち、

第  $i$  のハッシュ値は一方方向ハッシュ関数を  $m$  回適用したときの最小値であり、各適用は第  $i$  シード値と前記興味集合の  $m$  個の要素のそれぞれ 1 つをハッシュする

ことを特徴とする請求項 9 に記載の方法。

【請求項 11】

前記ユーザについての協調フィルタリングを実行するために、前記  $k$  個のクラスタを使用するステップをさらに有している

ことを特徴とする請求項 9 に記載の方法。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明はデジタルデータ処理に関し、より詳細にはコンピュータアプリケーションまたはシステムのユーザ (user) をクラスタ (cluster) にグループ分けすることに関する。

【背景技術】

【0002】

ユーザをクラスタにグループ分けすることは様々な目的で行われる。ユーザの個人性への適合化を達成するため、例えば、既知の技術の 1 つである協調フィルタリング (collaborative filtering) は、ユーザをクラスタリングすることおよびユーザにそのユーザクラスタ中の他のユーザが興味を示した品目 (item) を推奨することを含む。通常、例えば、ある品目をクリックした、それを購入した、またはそれを買い物カゴに入れた等の様々な様子から、ユーザはその品目に興味を示したとみなされてよい。推奨は様々な形をとることができ、例えば、ユーザに検索結果の一部として提示する、ユーザが読みたいかもしれないニュース記事として示す、ユーザが購入したいかもしれない品目を特定する、等々、がある。

30

【非特許文献 1】 J. Deans and S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, Proceedings of the 6th Symposium on Operating Systems Design and Implementation, pp.137-150 (December 6, 2004)

【非特許文献 2】 Charikar, Similarity Estimation Technique from Rounding Algorithms, 34th ACM Symposium on Theory of Computing, May 19-21, 2002, Montreal, Quebec, Canada

40

【発明の開示】

【発明が解決しようとする課題】

【0003】

ユーザのクラスタリングを達成する 1 つのやり方は、2 ユーザ間の距離測度を定義し、次に  $k$  平均法 ( $k$ -means) または階層的凝集型クラスタリング (hierarchical agglomerative clustering, HAC) 等の既知のクラスタリングアルゴリズムを使用して彼らをクラスタリングすることである。しかし、そのような技術には欠点がある。例えば、HAC は実行時間が  $O(n^2)$  であり、数億になる  $n$  の値については手に負えず、また  $k$  平均法ア

50

ルゴリズムではデータ点の平均値を提示する必要があるが、これはデータ点が集合の場合、不可能である。

【課題を解決するための手段】

【0004】

本発明は、特定の実施では、拡張性に富むユーザのクラスタリングを提供することができ、そこでは、各ユーザは品目母集団からの品目を表す要素からなる集合で表わされる。

【0005】

例えば、コンピュータシステムとの相互作用を通してユーザが選択できる品目の母集団が与えられたとき、各ユーザは品目の部分集合それぞれへの彼らの興味を様々な行動によって表現してもよく、それは、品目をクリックする、品目を購入する、品目を買い物リストに追加する、品目を眺める、等々である。本発明の特定の実施では、同じクラスタ中のユーザは彼らそれぞれの品目部分集合間に大きい重なりを有する可能性が大きいやり方で、ユーザをクラスタリングする（すなわち、ユーザをクラスタに割り当てる）。

【0006】

ある態様では、本発明の実施によるコンピュータプログラム製品は、データ処理装置に、複数ユーザの各々についてそれぞれの興味集合を取得させることができ、各興味集合はそれぞれのユーザがデータ処理システムとの相互作用を通して興味を示した品目を表わし、本製品はまた、データ処理装置に、複数ユーザの各々についてそれぞれの興味集合の  $k$  個のハッシュ (hash) 値を決定させることができ、第  $i$  のハッシュ値は対応する第  $i$  のハッシュ関数の下でのそれぞれの興味集合中の最小値であり、 $i$  は 1 と  $k$  の間の整数であり、 $k$  は 1 以上の整数であり、本製品はさらに、データ処理装置に、複数ユーザの各々をそれぞれのユーザについて第  $i$  のクラスタは第  $i$  のハッシュ値によって表わされるように確立されたそれぞれの  $k$  個のクラスタの各々に割り当てさせることができ、複数ユーザの各々の  $k$  個のクラスタへの割り当ては他のどのユーザについての  $k$  個のクラスタへの割り当てとも無関係に行われる。

【0007】

好適な実施は以下の特徴の 1 つ以上を含むことができる。本製品は、データ処理装置にユーザの興味を表わす動作をログに記録させるようにでき、また複数ユーザについての興味集合を生成するためそのログを使用させることができる。

【0008】

本製品は、データ処理装置に、複数ユーザ中の第 1 のユーザについて変化のあった興味集合を取得させることができ、第 1 のユーザについて変化のあった興味集合を使用して  $k$  個のハッシュ値を決定させることができ、そして第 1 のユーザを、変化のあった興味集合を使用して決定された  $k$  個のハッシュ値によって表わされるそれぞれの  $k$  個のクラスタの各々にのみ、他の複数ユーザのクラスタへの割り当てを变化させることなく、割り当てさせることができる。

【0009】

別の態様では、本発明の実施によるコンピュータプログラム製品は、データ処理装置に、あるユーザについて興味集合を取得させることができ、その興味集合はそのユーザがデータ処理システムとの相互作用を通して興味を示した品目を表わす。本製品はまた、データ処理装置に、興味集合の  $k$  個のハッシュ値を決定させることができ、第  $i$  のハッシュ値は対応する第  $i$  のハッシュ関数の下での興味集合中の最小値であり、 $i$  は 1 と  $k$  の間の整数であり、 $k$  は 1 以上の整数である。本製品はさらに、データ処理装置に、ユーザを  $k$  個のクラスタの各々に割り当てさせることができ、第  $i$  のクラスタは第  $i$  のハッシュ値により表わされる。

【0010】

好適な実施は以下の特徴の 1 つ以上を含むことができる。興味集合は  $m$  個の要素を持っており、第  $i$  のハッシュ値は一方向ハッシュ関数を  $m$  回適用したときの最小値であり、ここで、 $m$  回適用の各々は第  $i$  シード値 (seed value) と興味集合の  $m$  個の要素のそれぞれの 1 つをハッシュする。本製品は、ユーザのための協調フィルタリングを実行するために

10

20

30

40

50

、データ処理装置に k 個のユーザクラスタを使用させることができる。

【 0 0 1 1 】

別の態様では、本発明の実施によるシステムは、データ処理システムを使用する複数ユーザにより選択された品目のログと、複数ユーザの各々を k 個（ここで k は 1 以上の整数）のクラスタに割り当てるために指紋（fingerprint）関数および品目のログを使用する手段と、k 個のクラスタの 1 つ以上への第 1 のユーザの割り当てに基づいて複数ユーザ中の第 1 のユーザに情報を提供できる協調フィルタリングのコンピュータプログラムアプリケーションとを含む。

【 0 0 1 2 】

好適な実施は以下の特徴の 1 つ以上を含むことができる。その情報は推奨、予測、またはランク付けの少なくとも 1 つを含む。

【 0 0 1 3 】

別の態様では、本発明の実施によるコンピュータプログラム製品は、データ処理システムのユーザを同定するために、データ処理装置に k 個の要素からなる順序集合を使用させることができ、k は 1 より大きい整数であり、k 個の要素の各々は興味集合中の要素に対応し、興味集合中の各要素はデータ処理システムを使用するユーザによる行動を通してユーザが興味を示した品目を表す。

【 0 0 1 4 】

好適な実施は以下の特徴の 1 つ以上を含むことができる。本製品は、ユーザについての協調フィルタリングの実行においてユーザを同定するために、データ処理装置に k 個の要素からなる順序集合を使用させることができる。協調フィルタリングは、ユーザへの品目の推奨またはユーザへの品目のランク付けを含むことができる。本製品は、データ処理装置に、再編された興味集合を生成するためにデータ処理システムがその入力に応じて興味集合から要素を除去するような入力を、ユーザから受信させることができ、本製品はまた、データ処理装置に、k 個の要素からなる再編された順序集合を決定させることができ、k 個の要素の各々は再編された興味集合中の要素に対応しており、本製品はさらに、ユーザを同定するために、データ処理装置に、k 個の要素からなる初期の順序集合ではなく、k 個の要素からなる再編された順序集合を使用させることができる。k 個の要素からなる順序集合は、k 個のユーザクラスタの各々に属するものとしてユーザを同定する。本製品は、データ処理装置に、ユーザの興味を表す行動をログに記録させ、またユーザについての興味集合を生成するためにログを使用させることができる。データ処理システムはウェブサイトを含み、またユーザについての興味集合は、ユーザがウェブページでクリックした品目、ユーザがオンライン小売店から購入した品目、ユーザが買い物カートに追加した品目の 1 つ以上の表現を含む。品目への興味を表すユーザによる行動は、興味を暗に示す行動を含む。品目への興味を表すユーザによる行動は、興味を陽に示す行動を含む。ユーザはユーザのログオンによって同定される個人である。ユーザはクッキーによって同定される個人である。ユーザは観測される属性が共通する 1 人以上の個人であり、その属性は 1 人以上の個人の各々によりデータ処理システムに提示される属性である。ユーザはデータ処理システムと相互作用する個人のセッションである。興味集合中の各要素は、データ処理システムとの相互作用においてユーザが選択した品目である。

【 0 0 1 5 】

さらに別の態様では、本発明の実施は先述のプログラムおよびシステムに対応する方法、ならびに先述のシステムに対応するプログラムを含むことができる。

【 発明の効果 】

【 0 0 1 6 】

本発明は、以下の長所の 1 つ以上を実感するように実施することができる。クラスタリングの計算が拡張性に富む。計算は数億人の個々のユーザによって使用されるアプリケーションについて実行可能であり、個々のユーザは、数十、数百またはそれ以上の品目を彼らの興味集合中に表示させることができる。クラスタリングされたエンティティが品目の母集団の部分集合で表される場合、クラスタリングは実行可能である。母集団が予め定義

10

20

30

40

50

される必要は無い。クラスタリングは集合の類似性測度に基づく。新しいユーザのクラスタリングは既存のクラスタリングを変更することなく行われる。あるユーザのクラスタリングは、他のユーザがどうクラスタリングされたかまたはクラスタリングされつつあるかを考慮せずに行われる。しかし、例えば、シード値または順列 (permutation) 等の幾つかの広域的値は、クラスタリング間で共有されてもよい。例えば、品目の選択を実効的に廃棄または追加する等、選択を変更することにより、ユーザは彼らが割り当てられたクラスタを変更することができ、その場合、クラスタはその後計算または再計算される。新しいユーザまたは興味集合を修正したユーザがどのクラスタに属するかは、他のユーザからのデータを使用することなく計算可能である。クラスタリングの計算は、個人であるユーザのクラスタリングに限定されない。クラスタリングは、例えば、各ユーザが個人であるか、各ユーザが個人の集団であるか、各ユーザがシステムとの相互作用であるか、またはそれらの何らかの組合せであるかに関わらず、効率的に実行可能である。

10

【発明を実施するための最良の形態】

【0017】

本発明の1つ以上の実施形態について、添付の図面および以下の記述において説明する。本発明のその他の特徴および長所はそれらの記述、図面、および特許請求の範囲から明らかとなる。

【0018】

図面を通して、同じ参照番号および記号表示は同じ要素を示すものとする。

【0019】

20

図1は、ユーザをクラスタリングするミンハッシュ (minhash) 法に関する以下の論理的説明を図示したものである。本方法は実施可能であるが、ここでは説明の目的から原理を示すことにする。膨大な数のユーザを持つシステムにおいてユーザをクラスタリングする実際的な実施については後ほど図2を参照して説明されよう。

【0020】

図1に示すように、ミンハッシュ法への入力は、Uで示される品目母集団110と、 $p_1, p_2, \dots, p_k$ で示されるk個の順列からなる集合112と、ユーザAについて $X\_A$ で示されるユーザの興味集合114とである。

【0021】

順列はU上の順列であり、各順列は他の順列と同等に選択されるようU上の全ての順列の集合から均等に選ばれる。順列は各々UからUへの1対1写像 (全単射) である。そのような順列はUが固定しておりかつ可算である場合のみ実現可能である。整数kは選択可能なパラメータである。一般にkの値は5から10の範囲であろう。しかし、それは1またはそれ以上の任意の整数であってよい。本方法はユーザにk個のクラスタを割り当てよう。これらを $C_1, \dots, C_k$ で示す。順列が選択されユーザのクラスタへの割り当てに使用された後では、順列が変更された場合、全てのクラスタリングは再計算されなければならない。

30

【0022】

興味集合は母集団Uからの品目を表す要素からなる集合である。要素が品目それ自体である今説明中の用法では、興味集合は母集団Uからの品目のユーザによる選択の集合、 $X\_A$ 、である。これらは先述のようにして選択できる。本明細書での便宜のため、用語「品目 (item)」は、興味集合の要素かまたはユーザによる実際の選択かを指すことができるが、その意味は文脈から明らかであろう。

40

【0023】

このデータを使用して、ユーザについてk個、各順列について1個のハッシュ値が決定される (ステップ120)。順列 $p_i$ について、ハッシュ値を $h_i (X\_A)$ で示す。順列 $p_i$ についてのハッシュ値は、順列 $p_i$ の下での $X\_A$ からの最小要素、すなわちミンハッシュ値である。その最小値は、要素の値から、またはUの順序付けから決定することができる。

【0024】

50

各ミンハッシュ値はクラスタの識別子の役を果たし、ユーザはクラスタの各々へ割り当てられる。ユーザは  $k$  個のクラスタに属し、第  $i$  のクラスタは第  $i$  のミンハッシュ値によって同定される。このようにして、所与の順列  $p_i$  について、2つのユーザは、この順列の下での興味集合のミンハッシュ値が同じであり、かつその場合のみ、同じクラスタに属する。

#### 【0025】

各データ要素にハッシュ値を対応付けるこのミンハッシュ技術は、ローカリティセンシティブハッシング (locality sensitive hashing、局所性に敏感なハッシング) 技術と呼ばれるクラスの技術の1つであり、これは2つのデータ要素は2つのデータ要素間の類似性に直接比例するある確率で同じハッシュ値を有するという望ましい特性を有する。現在のケースでは、2人のユーザ  $A$ 、 $B$  (それらの興味集合  $X_A$ 、 $X_B$  で表される) 間の類似性は、( $X_A$  と  $X_B$  の積集合) の大きさを ( $X_A$  と  $X_B$  の和集合) の大きさを除したものと定義され、そこで、ミンハッシュ技術は、2人のユーザ  $A$  と  $B$  についてのミンハッシュ値が同じである確率 (使用される実際の順列がそこから選ばれる順列集合上で定義される) は、以上で定義した類似性測度に等しいという特性を有する。こうして、ミンハッシュ法は、ユーザ達は彼らの類似性に等しい確率で同じクラスタに落ち込むという、確率的クラスタリングを達成する。

#### 【0026】

$k$  個のクラスタが同定される (ステップ 122) ので、2人のユーザが同一クラスタ中にある確率が  $p$  ( $0 < p < 1$ ) の場合、あるクラスタリングでは彼らは同じクラスタに割り当てられなかったとしても、割合  $p$  のクラスタリングでは彼らは同じクラスタに割り当てられよう。これは、各ユーザは  $k$  個の異なるクラスタリングに均等に属し、各クラスタリングについては他の類似したユーザと同じクラスタに割り当てられるという平滑作用を与える。パラメータ  $k$  は効率 (小さい  $k$  はより高い効率を与える) と品質 (大きい  $k$  はより高い品質を与える) との間のトレードオフを最適化するように選択されるべきである。厳格に必然というのではないが、数値  $k$  は通常定数であって、10程度の小さい値でよい結果が得られる。

#### 【0027】

このミンハッシュクラスタリング法は、非常に拡張性に富んでおり、また幾つかの他の長所を有する。例えば、本方法の実行時間はデータの大きさ (すなわち、(user, item) 対 (pair) の総数) に比例する。

#### 【0028】

また、各ユーザは個々に、すなわち、他の全てのユーザとは独立にクラスタリングされる。これは四六時中ユーザが追加され、抹消され、そして更新されるウェブ領域においては特に興味深い。これに伴う長所は、従来のクラスタリングアルゴリズムでは困難な幾つかのケースを容易にまた追加的に処理できることである。ユーザがスパム的 (spammy) である、すなわち、クラスタリングを使用するシステムに影響を及ぼす目的で見せかけの興味を示していると判別された場合、そのユーザを他のどのユーザにも影響させずに、すなわち、残りのクラスタリングは不変のまま、抹消することができる。また、自分の選択を明かしてこなかったユーザがそれを明かそうと決心した場合、またはそのシステムに新しいユーザが追加された場合、他のユーザを再クラスタリングすることなく、そのユーザをクラスタに追加することができる。最後に、ユーザが自分のプロフィールを、自分の興味集合を事実上編集することにより変えようとして決心した場合、そのユーザについてのクラスタリングを、バッチ処理による更新とは対照的に、実時間で更新することができ、これを考慮に入れると、他のどのユーザのクラスタリングにも影響を与えない。

#### 【0029】

図2は、数億に上る膨大な数のユーザを有し、また各ユーザの興味集合中に、事実上または現実上可算でない品目母集団上で、数百以上の品目を有することもあるシステムにおいて、ユーザをクラスタリングする実際的な実施を示す。この実施はMapReduceプログラミングモデルおよび技術を使用するが、これについては後で説明する。

10

20

30

40

50

## 【 0 0 3 0 】

この実施への入力は、記号 D で示される、特別な秩序もなく蓄積されたデータ要素（例えば、結果のクリックログ、購入口ログ等）の寄集め（collection）2 1 0 と、 $s_1, s_2, \dots, s_k$  で示される  $k$  個のシード値からなる順序集合と、指紋関数 2 1 4 とである。各データ要素は、特定のユーザ（user）がある品目（item）に興味を示したことを示す（user, item）対と考えることができる。オプションとして、そのユーザがそのように行動した頻度を把握するために、そのデータ要素がそのユーザが興味を示した第 1、第 2 等々の事例に言及するかどうかを示すサフィックスを、品目の原形に付すことができる。好都合なことに、品目の形式はテキスト列であり、そのため、任意のウェブアプリケーションを通しての、すなわち、ユーザにユーザインタフェースを提示するウェブブラウザを使用する任意のアプリケーションを通しての、興味を表すどのようなユーザ行動も、そのような品目は容易に表現することができる。

10

## 【 0 0 3 1 】

興味を表すユーザ行動は、明白であることもあり、例えば、ユーザの興味を知らせる情報を、オンライン質問表への回答の形等でユーザがシステムに与えるとき、または暗黙的であることもある、例えば、ニュースサイトで読むニュース記事をユーザが選択するとき。

## 【 0 0 3 2 】

$k$  個のシード値  $s_1, s_2, \dots, s_k$  は、例えば、2 進表現でのビットが「0」、「1」均等で、ランダムに見えるように選ばれたビット列と考えられる数である。

20

## 【 0 0 3 3 】

指紋関数はシード値および（興味集合からの）品目を、例えば、6 4 ビットまたは 1 2 8 ビットの、大きい数に写像する。

## 【 0 0 3 4 】

ある実施では、シード値は U N I X（登録商標）の rand 関数を使用して生成され、 $k$  個の 3 2 ビット整数値を生成する。rand 関数は、1 個のシード値の生成に 1 度以上呼び出さなければならないかもしれない。この実施では、指紋関数は M D 5 一方向ハッシュアルゴリズムを実装しており、品目（これは一般にテキスト列または 2 進データであろう）と連結したシード値をハッシュして 1 2 8 ビットの値を作る。

## 【 0 0 3 5 】

シード値と指紋関数は、図 1 を参照して説明した  $k$  個の順列  $p_1, \dots, p_k$  に論理的に対応しており、品目の可算母集団を必要とせずに、品目の順序付けと順列を提供する。

30

## 【 0 0 3 6 】

寄集め D はマプリアデュース（MapReduce）の枠組みを使用して処理される。これについては後で説明する。

## 【 0 0 3 7 】

map フェーズ 2 2 0 では、各（user, item）対について、key=user かつ value=item とした（key, value）対が分散的に出力される。

## 【 0 0 3 8 】

reduce フェーズ 2 2 2 では、同じ key(user) を有するような全ての（key, value）対が集められて、reduce ルーチンに与えられ、これは各々の個別の key(user) について 1 度、分散的に実行される。

40

## 【 0 0 3 9 】

（特定のユーザについて）reduce ルーチンはそのユーザの興味集合中の全ての品目を処理する。この説明のため、これらの  $m$  個の品目を  $i_1, i_2, \dots, i_m$  で示す。各シード値  $s_i$  について、reduce ルーチンは、品目とシード値の指紋、すなわち、 $\text{fingerprint}(s_i, i_l)$ 、である  $m$  個（各品目について 1 つ）の値を計算する。 $m$  個の品目についてのこれらの  $\text{fingerprint}$  の最小値が計算され、それが第  $i$  のシード  $s_i$  に対応する第  $i$  のミンハッシュ値となる。

## 【 0 0 4 0 】

50



ユーザは、このようにして算出されたk個のミンハッシュ値によって表わされる。これらは、そのユーザが属するk個のクラスタを表現し、そのユーザはこれらのクラスタに割り当てられたと称される。

【0041】

図3に示すように、推奨者コンピュータプログラムアプリケーションは、本明細書で説明される任意の方法で生成されるユーザクラスタを使用してよい。

【0042】

ある実施では、システムはユーザによってなされた選択をログに記録する(ステップ310)。そのログは、ディスク駆動装置またはファイルサーバに、例えば、無構造のテキスト行または構造化されたデータベース中のレコード等、どのような形式で蓄積されてもよい。システムは、検索結果、広告、購買選択、そのサイトの内部または外部のページへの単純なリンク、またはその他の品目、をサービスするウェブサイトであってよい。ログされた選択は、そのシステムのユーザによってなされた全ての選択であってよいが、そうである必要もない。例えば、アプリケーションは全サイトというよりはニュースサイトのみを選択に、または示された全品目というより購買目的の品目のみの選択に、興味を持っているのかもしれない。さらに、システムは、種々の推奨者アプリケーション用の異なる種類の複数の選択ログを保持することができ、推奨者アプリケーションはそれら自身のそれぞれのユーザのクラスタリングを計算できる。例えば、シードおよびfingerprint関数を使用する方法では、別個のクラスタリングの各々はそれ自身の別個のシード列およびfingerprint関数を持つことができる。

【0043】

システムは、ユーザ登録とログオンにより、クッキーによるかまたは他のやり方で個人をユーザとして判別可能である。オプションとして、もし個々のユーザに関する情報をシステムとの相互作用の複数セッションに跨って保持することが望ましくない場合は、システムは、ユーザセッションをクラスタリング目的用のユーザとして扱うことができる。クッキーをセッションの維持に使用することもできる。(クッキーは、サーバによってウェブブラウザへ送られ、その後ブラウザによってサーバへのアクセスの都度送り戻されるパケット型の情報である。)オプションとして、システムは、ロギングに参加するか否か、すなわち、自分自身を選択のロギングに含めるか除くかを、個人が決定できるようにできる。

【0044】

オプションとして、システムは、ユーザとしてシステムと相互作用する個人のある属性または属性の組合せを処理することができる。属性はシステムによって観測できてよく、例えば、使用されるIP(インターネットプロトコル)アドレスまたは使用される言語、または個人が提供する情報、例えば、住所の都市または国、またはシステムが提供するサービスへの加入であってよい。こうして、例えば、システムはクパーティノからの個人を一ユーザとして、またレドモンドからの個人を別のユーザとして処理することもできよう。そのような寄集め型のクラスタリングの利点は、ロギンまたは登録を要求することなく、システムのある程度の個人性への適合化が可能になることである。さらに、オプションとして、システムは全ての種類のユーザ、例えば、個人または集団を、同じクラスタと一緒にクラスタリングすることもでき、または別の種類のユーザについては別のクラスタを確立することもできる。

【0045】

システムのユーザが行う選択は、単純な選択であることができ、またはオプションとして、複合的な選択であることができる。複合的な選択は一連の選択であり、例えば、第1のウェブページへナビゲート(navigate)し、それから直接に第2のウェブページへナビゲートする一連のナビゲーションである。ウェブページはリソース、通常はHTML(Hypertext Markup Language、ハイパーテキストマークアップ言語)文書、であり、ウェブサーバによってウェブブラウザにサービスされる。ウェブサーバは、通常はネットワークを介して受信されるHTTP(Hypertext Transfer Protocol、ハイパーテキスト転送プ

10

20

30

40

50

ロトコル) 要求を受け付けるコンピュータプログラムであり、要求者に対してHTTP 応答を与える。HTTP 応答は通常HTML 文書を含むが、テキストファイル、動画、または何らかのその他のタイプの文書であってもよい。

【0046】

ログされた選択に基づいて、ユーザは各々k 個のクラスタに割り当てられる(ステップ312) が、これについては本明細書の別の箇所で説明されている。このユーザのクラスタリングは、システムに新しいユーザが現れたとき、および選択がログに追加または除去されたときに、更新することができる。オプションとして、ある状況では必ずしも全てのユーザがk 個のクラスタに割り当てられるとは限らない。そのような状況では、特定のユーザへの推奨を見つけるために、1 個以上でk 個より少ないクラスタ識別子を取得することができる。例えば、一連の選択を持つ新しいユーザに推奨を提供する要求をシステムが受けると、オプションとして、システムはそれらの選択を使用して第1 のクラスタの識別子を計算し、推奨を見つけるためにそれを使用し、続けて第2 のクラスタを同様に計算して使用し、等々、システム定義による十分な数の推奨が見つかるまで続けることができる。

10

【0047】

推奨者アプリケーションは次いで、特定ユーザへの推奨を行うためにそのユーザのクラスタを使用することができる(ステップ314)。各ユーザの単一クラスタへのグループ分けに基づいて推奨を行うどのような方法も、本明細書で説明された複数クラスタとともに使用することができる。例えば、そのような方法はk 回適用可能であり、そのユーザについての推奨品目の和集合を得るためにk 個の結果がマージされる。あるいは、ある品目が出現する結果の数をその品目のランク付けに使用することができる。または、ユーザに多様な推奨を提供するために、クラスタに基づく推奨結果の各々から若干ずつの品目をそのユーザに提供することができる。ユーザが割り当てられた複数のクラスタは、システムを使用する際にユーザが持った様々な種類の興味を反映しているかもしれない、そこでユーザにそのような多様な推奨を与えることは、単一のクラスタが使用された場合より、推奨がユーザのそのときの興味に触れる何かを含む可能性を高めることになる。

20

【0048】

推奨者アプリケーションは協調フィルタリングの一例であり、本明細書で説明されたユーザのクラスタリングの方法は別の種類の協調フィルタリングにも同様に適用することができる。協調フィルタリングでは現ユーザに類似のユーザが見つかり、かれらの嗜好または行動から現ユーザ向けのランク付け、推奨または予測が行われる。ユーザを複数のクラスタへグループ分けすることにより、システムは暗黙裡にユーザの嗜好を判別し、ユーザのグループ分けを通して品目をグループ分けする。

30

【0049】

図4 に概略的に示されるように、本明細書で説明されたユーザをクラスタに割り当てる技術は、ユーザ402a、402b に提示されるニュース記事の推奨を、それらのユーザによって以前になされた記事の選択に基づいて提供することができるニュース推奨エンジン410 において実施することができる。ユーザ402a、402b は、彼等それぞれのブラウザを通し、例えば、ローカル、広域、または仮想プライベートネットワーク、もしくはインターネット等の通信ネットワーク404 を介して、1 つ以上のウェブサーバ430 と通信する。ニュースサービス420 は、サーバ(1 つまたは複数) 430 に収容されたコンピュータプログラムとして実施され、ユーザの要求に応じてユーザ402a、402b にウェブページをサービスする。ニュースサービス420 によってサービスされるページには、ユーザが1 つまたは複数のニュースを選択してユーザのブラウザで表示できるページが存在する。ユーザの選択に応じて、ニュースサービス420 はユーザが選択した記事をサービスする(機能424)。ニュース推奨エンジン410 が特定ユーザについて推奨を提供した場合、そのユーザについての推奨に応じて、ニュースサービスはそのユーザによる選択のための記事を表示するページをサービスすることができる(機能422)。

40

50

## 【 0 0 5 0 】

ニュース推奨エンジン 4 1 0 はサーバ ( 1 つまたは複数 ) 4 3 0 で実行されるコンピュータプログラムとして実施される。ニュース推奨エンジン 4 1 0 はニュースサービス 4 2 0 のユーザから選択を受信し、それらの選択をログ 4 4 0 にログする ( 機能 4 1 2 )。本明細書の別の箇所での説明のように、ログ 4 4 0 中の情報を使用して、エンジンはユーザをクラスタに割り当て ( 機能 4 1 4 )。クラスタに割り当てられたいずれの特定のユーザについても、エンジンはそのユーザが割り当てられたクラスタに基づいて推奨を決定し ( 機能 4 1 6 )、それらの推奨をニュースサービス 4 2 0 に提供する。

## 【 0 0 5 1 】

特定のユーザについてなされる推奨の決定に際して、エンジンはその特定ユーザと同じクラスタ ( 1 つまたは複数 ) に割り当てられた他のユーザによってなされた選択を考慮する。可能性のある推奨の中から、オプションとして、エンジンはユーザが既に選択したニュース記事を消去することができる。エンジンまたはサービスは、あるニュース記事がそのユーザが割り当てられたクラスタに割り当てられている他のユーザによって選択された回数、そのニュース記事がどの程度新しいものか、問題のニュース記事の主題に関する記事を有するソースの数、等の種々の基準に基づいて推奨をランク付けすることができる。このようにして、ニュースサービスはそのユーザに対して個人性に適合化されたニュース記事の提示およびランク付けを提供することができる。

10

## 【 0 0 5 2 】

ある実施では、ニュース推奨エンジン 4 1 0 はユーザを個人として同定しており、そこで個人性に適合化した推奨を得るべく、ユーザにログインし登録するよう要求する。別の実施では、本明細書の別の箇所での説明のように、ユーザを暗黙裡にまたは寄集めグループとして同定することができる。

20

## 【 0 0 5 3 】

その他の種類のサービス、例えば、動画、ブログ、または買い物情報等の選択を提供するサービスの個人性への適合化をサポートするために、推奨エンジンをこの線に沿って実施することができる。

## 【 0 0 5 4 】

図 4 では別モジュールとして示されているが、エンジンおよびサービスの機能はこのようなやり方で実施される必要はなく、特に、エンジンはサービスの実施の一部として実施することができる。

30

## 【 0 0 5 5 】

以下の段落では、MapReduce プログラミングモデルおよび大きいデータ集合を処理し生成するためのモデルの実装について説明する。モデルおよびそのライブラリの実装とともに MapReduce と呼ぶものとする。MapReduce を使用して、プログラマは、key/value 対を処理して中間 key/value 対の集合を生成する map 関数、および同じ中間 key に対応する全ての中間 value を 1 つにまとめる reduce 関数、の仕様を定める。この関数スタイルで作成されたプログラムは自動的に並列化され、日用品的コンピュータの大集団 ( cluster ) 上で実行することができる。ランタイムシステムまたはフレームワークは、入力データを分割し、一連のマシン全域でのプログラム実行をスケジュールし、マシン障害を処理し、そしてマシン間で必要な通信を管理するように実装することができる。

40

## 【 0 0 5 6 】

MapReduce 計算は一連の入力 key/value 対を取り込み、一連の出力 key/value 対を生成する。ユーザはその計算を 2 つの関数、map と reduce で表現する。

## 【 0 0 5 7 】

map、ユーザにより作成、は入力 key/value 対を取り込み、一連の中間 key/value 対を生成する。MapReduce ライブラリは同じ中間 key *l* に対応する全ての中間 value をまとめてグループ化し、それらを reduce 関数に渡す。

## 【 0 0 5 8 】

reduce 関数、これもユーザにより作成、は中間 key *l* とその key に対応する一連の value

50

を受け取る。それはこれらのvalueを一緒にマージして、可能性としてはより小さなvalue集合を形成する。通常はreduceを呼ぶ度に0または1個の出力valueが生成される。中間valueはイテレータ(iterator)を通してユーザのreduce関数に供給される。このようにして、大き過ぎてメモリに収まらないvalueのリストも手に負えるようになる。

【0059】

文書の大量集積について各語の生起回数を計数する問題を考えよう。ユーザは以下の擬似コードに似たコードを作成することになる。

```
map (String key, String value):
```

```
    // key: document name
```

```
    // value: document contents
```

```
    for each word w in value:
```

```
        EmitIntermediate (w, "1") ;
```

```
reduce (String key, Iterator values):
```

```
    // key: a word
```

```
    // values: a list of counts
```

```
    int result = 0;
```

```
    for each v in values:
```

```
        result += ParseInt (v) ;
```

```
    Emit (AsString (result) ) ;
```

【0060】

map関数は各語および対応する生起回数(この簡単な例では只の「1」)を併せて出力する。reduce関数は特定の語について出力された全ての回数を足し合わせる。

【0061】

ある実施では、計算を実行するために、ユーザは入力ファイルおよび出力ファイルの名前とともに仕様目的ならびにオプションの調整用パラメータを記入するコードを作成する。ユーザはそこでMapReduce関数を呼び出し、それに仕様目的を渡す。ユーザのコードはMapReduceライブラリと併せてリンクされる。

【0062】

上記の擬似コードは入出力が文字列であるとして書かれたものであるが、概念上はユーザが提供するmap関数およびreduce関数は関連したタイプを有しており、

```
map(k1, v1) list(k2, v2)
```

```
reduce(k2, list(v2)) list(v2)
```

すなわち、入力keyおよび入力valueは、出力keyおよび出力valueとは別のドメインから引き出される。さらに、中間keyおよび中間valueは、出力keyおよび出力valueと同じドメインから引き出される。

【0063】

MapReduceモデルについては多くの異なった実装が可能である。

【0064】

以下の段落では、スイッチ機能付きイーサネット(登録商標)で相互接続された日用品的パソコンの大集団からなる計算環境を目標とする実装について説明する。この環境では、マシンはマシン当たり通常2から4GB(ギガバイト)のメモリを有し、集団は数百または数千のマシンを有し、外部メモリは個々のマシンに装着された低廉なIDE(Integrated Drive Electronics standard)ディスクにより与えられ、これらのディスクは低信頼度のハードウェア上で可用性および信頼性を備えるために複製を使用するが、そこに蓄積されたデータを管理するために分散型のファイルシステムが使用され、ユーザはスケジューリングシステムにジョブを提出する。各ジョブは一連の仕事からなり、スケジューリングシステムのスケジューラによって、集団中の一連の利用可能なマシンへと計画的に割り振られる。

【0065】

mapの呼び出しは、入力データを一連のM個のスプリット(split)へ自動的に分割する

ことにより、複数マシンにわたって分散化される。入力スプリットを異なるマシンにより並行処理することができる。reduceの呼び出しは、分割関数（例えば、 $\text{hash}(\text{key}) \bmod R$ ）を使用して中間key空間をR個に分割することにより分散化される。分割数（R）および分割関数はユーザが指定する。

【0066】

ユーザプログラムがMapReduce関数を呼ぶと、以下の一連の動作が行われる。

【0067】

1. ユーザプログラム中のMapReduceライブラリは、まず、入力ファイルを1個当たり（ユーザが制御可能な）通常16メガバイトから64メガバイト（MB）のM個へと分割する。それから、プログラムのコピー多数をマシン集団上で起動する。

10

【0068】

2. プログラムのコピーの内の1つはマスタ（master）である。残りは、マスタ（master）によって作業を割り付けられるワーカー（worker）である。割り付けるべきM個のmapタスクとR個のreduceタスクが存在する。マスタはアイドル状態（idle）のワーカーを選び、各々にmapタスクまたはreduceタスクを割り付ける。

【0069】

3. mapタスクを割り付けられたワーカーは対応する入力スプリットの内容を読む。それは入力データからkey/value対を構文解析し、各対をユーザ定義のmap関数へ渡す。map関数によって生成された中間key/value対はメモリにバッファされる。

【0070】

20

4. 周期的に、バッファされた対は、分割関数によってR個の領域に分割（partition）されたローカルディスクへ書き込まれる。これらのバッファされた対のローカルディスク上の位置はマスタに戻され、マスタはこれらの位置をreduceワーカーへ責任をもって転送する。

【0071】

5. reduceワーカーがマスタによってこれらの位置について知らされると、それは遠隔手順呼出しを使用して、バッファされたデータをmapワーカーのローカルディスクから読み取る。全ての中間データを読み取ると、reduceワーカーはそれを中間keyによってソート（sort）し、同一keyの全オカレンス（occurrence）をグループにまとめる。通常多くの異なるkeyが同一のreduceタスクに割り振られるので、このソートが役に立つ。中間データの量が多過ぎてメモリに収まらない場合は、外付けのソートが使用される。

30

【0072】

6. reduceワーカーはソートされた中間データに対して繰り返し動作し、目新しい中間keyに出会う都度、そのkeyおよび対応する中間valueの集合をユーザのreduce関数に渡す。reduce関数の出力はこのreduceパーティションについての最終出力ファイルに付加される。

【0073】

7. 全てのmapタスクおよびreduceタスクが完了すると、マスタはユーザプログラムを呼び起こす。この時点で、ユーザプログラム中でのMapReduce呼出しはユーザコードへ戻る。

40

【0074】

成功裏に完了した後、実行出力はR個の出力ファイル（reduceタスク当たり1個、ユーザが指定したファイル名で）で利用可能となる。ユーザはこれらのR個の出力ファイルを1個のファイルに結合する必要はなく、これらのファイルを別のMapReduce呼び出しへの入力として渡すか、またはそれらを複数ファイルに分割された入力を処理可能な他の分散アプリケーションから使用することができる。

【0075】

マスタは幾つかのデータ構造を保持する。各々のmapタスクおよびreduceタスクについて、マスタは状態（アイドル、進行中、または完了）および（非アイドルタスクについての）ワーカーマシンの識別子を蓄積する。

50

## 【 0 0 7 6 】

マスタは、中間ファイル領域の位置をmapタスクからreduceタスクへ伝えるパイプの役を果たす。したがって、完了したmapタスクの各々について、マスタはそのmapタスクによって生成されたR個の中間ファイル領域の位置と大きさを蓄積する。この位置と大きさの情報への更新は、mapタスクが完了したときに受信される。その情報は進行中reduceタスクを有するワーカに追加的に押し込まれる。

## 【 0 0 7 7 】

このMapReduceライブラリの実装は数百または数千のマシンを使用して膨大な量のデータを処理するように設計されるので、このライブラリはマシン故障にも耐用性がある。

10

## 【 0 0 7 8 】

マスタはどのワーカに対しても周期的に接続確認信号を発する (ping)。ある時間内にワーカから返答が受信されない場合、マスタはそのワーカに故障の印を付ける。そのワーカが完了したどのmapタスクもリセットされて初期のアイドル状態に戻され、したがって他のワーカ上でスケジューリングできるようになる。同様に、故障したワーカ上で進行中のどのmapタスクまたはreduceタスクもアイドル状態にリセットされ、再スケジュールできるようになる。

## 【 0 0 7 9 】

故障時には完了したmapタスクは再実行される。というのは、それらの出力が故障マシンのローカルディスクに蓄積されており、アクセスできないからである。完了したreduceタスクは、それらの出力がグローバルファイルシステムに蓄積されるので再実行の必要はない。

20

## 【 0 0 8 0 】

Mapタスクが初めにワーカAによって実行され、その後、(ワーカAが故障したため)ワーカBによって実行された場合、reduceタスクを実行する全てのワーカにその再実行が知らされる。ワーカAからのデータ読み出しを実行済みでない全てのreduceタスクは、ワーカBからデータを読み出すことになる。

## 【 0 0 8 1 】

マスタは1つしか存在しないのでその故障は滅多にない。したがって、マスタが故障した場合、MapReduce計算は打ち切りとなる。ユーザまたはユーザプログラムは、この状況をチェックすることができ、かれらが望む場合、MapReduceの動作を再試行する。

30

## 【 0 0 8 2 】

ユーザ提供のmapおよびreduceオペレータ (operator) がそれらへの入力値の決定性関数である場合、この分散の実装は、全プログラムを故障なしで順番に実行することによって生成されるのと同じ出力を生成する。各進行中タスクはその出力を自分だけの一時ファイルへ書き出す。mapタスクが完了すると、ワーカはマスタにメッセージを送信し、そのメッセージ中にR個の一時ファイルの名前を含ませる。マスタが既に完了したmapタスクについて完了メッセージを受信した場合、そのメッセージを無視する。そうでない場合、マスタはR個の一時ファイルの名前をマスタのデータ構造中に記録する。reduceタスクが完了すると、reduceワーカはそれの一時出力ファイルを最終出力ファイルに原子レベルで名前変更する。同じreduceタスクが複数のマシンで実行される場合、同じ最終出力ファイルについて複数の名前変更要求が実行されることになる。基盤的なファイルシステムによって提供される原子レベルの名前変更動作は、最終ファイルシステムの状態がreduceタスクの1回の実行によって生成されるデータだけを含むことを保証する。

40

## 【 0 0 8 3 】

本実装は、集団を構成するマシンのローカルディスクに入力データを蓄積するという事実を活用して、ネットワーク帯域幅の浪費を防いでいる。ファイルシステムは各ファイルを64MBブロックに分割し、各ブロックのコピーを別々のマシンに蓄積する。MapReduceマスタは、入力ファイルの位置情報を考慮にいて、対応する入力データの複製を含むマシン上のmapタスクのスケジュールを試みる。それに失敗すると、そのタスクの

50

入力データの複製に近い(すなわち、そのデータを含むマシンと同じネットワークスイッチ上にあるワーカマシン上の)mapタスクのスケジュールを試みる。

【0084】

負荷を動的に均衡させるためには、MおよびRはワーカマシンの数より遥かに大きくあるべきである。先に述べたように、マスタは $O(M+R)$ のスケジュール判断を実行し、 $O(M \times R)$ の状態をメモリ中に保持しなければならないので、この実装においてMおよびRをどの程度に大きくできるかには実用上の限界がある。さらに、各reduceタスクの出力は別々の出力ファイル中で閉じているので、Rはユーザによってしばしば制限される。実用上は、Mは、先述の局地最適性が最も効果的となるよう個々のタスクの各々が最大に16MBから64MBの入力ファイルを有するように選ばれ、Rは使用を期待できるワーカマシン数の小さい倍数となろう。

10

【0085】

MapReduceの実行にかかる総時間は、落ちこぼれ、すなわち、計算における最後の幾つかのmapまたはreduceタスクの1つを完了するのに異常に長時間を要するマシン、によって大きく影響され得る。落ちこぼれ問題を軽減するために、MapReduceの実行が完了に近づくと、マスタはまだ進行中のタスクの支援実行をスケジュールする。本来の実行かまたは支援実行かが完了する都度、タスクは完了の印が付けられる。

【0086】

以上の基本機能に加えて、本実装は以下の有用な拡張機能を提供する。

【0087】

幾つかのケースでは、keyの何か特定の機能によりデータを分割するのが有用である。これをサポートするために、MapReduceライブラリのユーザは分割関数を提供することができる。

20

【0088】

本実装は、所与のパーティション内では、中間key/value対がkeyの昇順に処理されることを保証する。これは、パーティション毎のソートされた出力ファイルの生成を容易化して、出力ファイルの形式がkeyによる効率的なランダムアクセス検索サポートする必要がある場合、または出力のユーザにとってデータがソートされていると好都合な場合に役に立つ。

【0089】

幾つかのケースでは、各mapタスクによって生成される中間keyにおける繰り返しが膨大で、そしてユーザ仕様によるreduce関数が可換でありかつ結合法則が成立する。これの例は先述の語数を計数する例である。各mapタスクは<the, 1>の形式の数百または数千のレコードを生成する場合もある。これらの計数値は全て、ネットワークを介して単一のreduceタスクへ、そしてreduce関数によって総和を取られ1つの数を生成するために、送信される。そのようなケースに備えるため、本実装は、オプションとして、ネットワークを介してデータを送信する前にそれを部分的にマージするcombiner関数をユーザが指定できるようにする。

30

【0090】

combiner関数は、mapタスクを実行する各マシン上で実行される。combiner関数およびreduce関数の両者の実装に同じコードを使用できる。reduce関数とcombiner関数との間の唯一の違いは、MapReduceライブラリが関数の出力をどう処理するかである。reduce関数の出力は最終出力ファイルに書き出される。combiner関数の出力は、reduceタスクへ送信される中間ファイルに書き出される。

40

【0091】

MapReduceライブラリは、複数の異なったフォーマットの入力データの読み取りをサポートする。例えば、「テキスト」モードの入力は各行をkey/value対として処理し、ここで、keyはファイル中のオフセット(offset)であり、valueは行の内容である。もう1つの一般的にサポートされるフォーマットは、keyでソートされた一連のkey/value対を蓄積する。入力タイプの各実装は、そのタイプのデータを、個別のmapタスクとして処理する

50

意味がある領域に分割する方法を知っている（例えば、テキストモードの領域分割では、領域の分割は、確実に行の境界でのみ行われる）。ユーザは、簡単なreaderインタフェースの実装を提供することにより、新しい入力タイプへのサポートを追加できる。さらに、readerはファイルから読み出されたデータの提供に限定されない。例えば、readerはデータベースまたはメモリ中に割り振られたデータ構造からレコードを読み出すことができる。

【 0 0 9 2 】

同様のやり方で、本実装は様々なフォーマットでデータを生成する一連の出力タイプをサポートし、またユーザが新しい出力タイプのサポートを追加するためにコードを書くことは容易である。

10

【 0 0 9 3 】

ユーザまたは第3者のコード中のバグにより、map関数またはreduce関数があるレコードで決定的に機能停止することも時にはある。例えば、大きいデータ集合の統計的解析を行っているとき等、若干のレコードを無視することが受容可能な場合もある。本実装は、MapReduceライブラリが決定的機能停止を引き起こしたレコードを検出し、前進するためにこれらのレコードをスキップする、オプションとしての実行モードを提供する。

【 0 0 9 4 】

このモードのために、各ワーカのプロセスは、セグメンテーション違反およびバス誤りを捕捉する信号ハンドラ(handler)を組み込んでいる。ユーザのmapまたはreduceの実行を呼び出す前に、MapReduceライブラリは引数の順序番号をグローバル変数中に蓄積する。ユーザコードが信号を発生する場合、信号ハンドラはその順序番号を含む「最後のあえぎ(last gasp)」UDP(User Datagram Protocol)パケットをMapReduceマスタに送信する。特定のレコードについて1つ以上の故障を見つけると、マスタは対応するmapまたはreduceタスクの次の再実行の指令発出に際して、そのレコードをスキップ(skip)すべきことを知らせる。

20

【 0 0 9 5 】

MapReduceに関するさらに多くの情報は、その内容が参照によって本明細書に組み込まれた非特許文献1で見つけることができる。

【 0 0 9 6 】

ローカリティセンシティブハッシュ体系を使用してユーザを複数のクラスタにクラスタリングするもう1つの方法についてここで簡単に説明しよう。この方法では、各ユーザは、ユーザを特徴付ける高次元のベクトルで表現されたプロフィールを有する。そのようなベクトル上で動作する一連のk個のハッシュ関数が選択される。ユーザプロフィールについての第iのハッシュ値はユーザが割り当てられる第iのクラスタを表す。本方法に有用なローカリティセンシティブハッシュ関数については非特許文献2に説明がある。

30

【 0 0 9 7 】

そのような方法のある実装では、ユーザは<term, weight>対のリストにより表現される。先述と同様、kはクラスタの数およびユーザに対して計算されるハッシュ値の数である。シード値の数は説明では8kで与えられようが、定数で与えられている8は一般にはパラメータである。8k個のランダムなシード値は、s<sub>1</sub>, s<sub>2</sub>, ..., s<sub>8k</sub>で示す文字列で表現され、例えば、2進表現でのビットの「0」または「1」が均等で、ランダムに見えるように選択される。どのユーザについても、第iのハッシュ値は以下のように計算される。

40

【 0 0 9 8 】

```

For b from 1 to 8:
do
  initialize sum = 0;
  for all <term_j, weight_j> pairs in the user's list:
do
  if (fingerprint (term_j + s_((i-1)*8 + b)) has least significant bit = 1

```

50



```

)
    sum = sum + weight_j
else
    sum = sum - weight_j
done
if (sum > 0)
    b-th bit of i-th hash value is set to 1.
else
    b-th bit of i-th hash value is set to 0.
done.

```

10

## 【0099】

fingerprint (term<sub>j</sub> + s<sub>((i-1)\*8 + b)</sub>)なる項は、シード文字列s<sub>((i-1)\*8 + b)</sub>、すなわち第((i-1)\*8 + b)のシード文字列と連結された第j項(term<sub>j</sub>)のfingerprint関数(先述のように計算される)を表わす。

## 【0100】

本発明の実施形態および本明細書で説明した全ての機能的動作は、本明細書で開示された構造またはそれらの構造的等価物もしくはそれらの組合せを含む、デジタル電子回路、もしくはコンピュータソフトウェア、ファームウェア、またはハードウェアで実施可能である。本発明の実施形態は、1つ以上のコンピュータプログラム製品、すなわち、例えば、マシン可読蓄積装置、マシン可読蓄積媒体、メモリ装置、またはマシン可読伝播信号、等のコンピュータ可読媒体上にコード化された、データ処理装置による実行に使用されるか、またはその動作を制御する、コンピュータプログラム命令の1つ以上のモジュールとして実施可能である。用語「データ処理装置」は、例として、プログラム可能プロセッサ、コンピュータ、もしくは多重プロセッサまたは多重コンピュータを含む、データを処理するための全ての装置およびマシンを包含する。装置は、ハードウェアに加えて、問題になっているコンピュータプログラムのための実行環境を創り出すコード、例えば、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、またはそれらの組合せを構成するコード、を含む。伝播信号は、例えば、機械により発生された電氣的、光学的または電磁的的信号等の人工的に発生された信号であり、好適な受信装置への転送用に情報を符号化するために発生される。

20

30

## 【0101】

コンピュータプログラム(プログラム、ソフトウェア、ソフトウェアアプリケーション、スクリプト、またはコードとしても知られる)は、コンパイルまたはインタープリタ言語を含むどのような形式のプログラミング言語でも書くことができ、スタンドアロンプログラムとしてもしくはモジュール、コンポーネント、サブルーチン、またはコンピュータ環境での使用に好適なその他の単位を含む任意の形式で配備できる。コンピュータプログラムは必ずしもファイルシステム中のファイルに対応しなくてもよい。プログラムは、他のプログラムまたはデータを保持するファイルの一部に(例えば、マークアップ言語文書中に蓄積された1つ以上のスクリプト)、問題になっているプログラムに専用の単一のファイルに、または複数の組織的なファイル(例えば、1つ以上のモジュール、サブプログラム、またはコードの一部を蓄積するファイル)に、蓄積することができる。コンピュータプログラムは、1つのコンピュータで、もしくは1つのサイトに配置された、または複数サイトにわたって分散され、通信ネットワークによって相互接続された、複数のコンピュータで、実行するために配備することができる。

40

## 【0102】

本明細書で説明したプロセスおよび論理の流れは、入力データに対して動作して出力を生成することにより機能を実行する1つ以上のコンピュータプログラムを実行する、1つ以上のプログラム可能なプロセッサによって実行することができる。例えば、FPGA(Field Programmable Gate Array、フィールドプログラマブルゲートアレイ)またはASIC(Application-Specific Integrated Circuit、特定用途向け集積回路)等の特定用

50

途論理回路により、そのプロセスおよび論理の流れを実行することもでき、装置を実装することもできる。

【0103】

コンピュータプログラムの実行に好適なプロセッサには、例として、汎用および特定目的用の両方のマイクロプロセッサ、ならびに任意の種類のデジタルコンピュータの任意の1つ以上のプロセッサが含まれる。一般に、プロセッサは読み出し専用メモリまたはランダムアクセスメモリもしくはその両方から命令とデータを受け取るであろう。コンピュータの本質的要素は、命令を実行するプロセッサおよび命令とデータを蓄積する1つ以上のメモリである。コンピュータはまた、一般に、データを蓄積するための1つ以上の大容量記憶装置、例えば、磁気、磁気光学ディスク、または光学ディスク、を含むか、またはそれらからデータを受信しまたはそれらへデータを送信しもしくはその両方を実行できるようにそれらに接続されよう。しかし、コンピュータはそのような装置を持たなくてもよい。さらに、コンピュータは別の装置、幾つか名前を上げれば、例えば、携帯電話機、携帯情報端末(PDA、personal digital assistant)、携帯オーディオプレーヤ、全地球測位システム(GPS、Global Positioning System)受信機、等に埋め込むこともできる。コンピュータプログラム命令およびデータを具体化するのに好適な情報担体には、全ての形態の不揮発性メモリ、例としては、EPROM、EEPROM、およびフラッシュメモリ装置等の半導体メモリ装置、内蔵ハードディスクまたは取り外し可能ディスク等の磁気ディスク装置、磁気光学ディスク、およびCD-ROMおよびDVD-ROMディスク、が含まれる。プロセッサおよびメモリは特定用途論理回路で補足することができ、またはそれに組み込むことができる。

10

20

【0104】

ユーザとの相互作用に備えて、本発明の実施形態は、情報をユーザに提示するための表示装置、例えば、CRT(cathode ray tube、陰極線管)またはLCD(liquid crystal display、液晶ディスプレイ)モニタ、ならびにユーザがコンピュータに入力を与えることができるキーボードおよび指示装置、例えば、マウスまたはトラックボール、を有するコンピュータ上で実施することができる。その他の種類の装置もまたユーザとの相互作用に備えて使用することができ、例えば、ユーザに提供されるフィードバックは、例えば、視覚的フィードバック、聴覚的フィードバック、または触覚的フィードバック等の任意の形式の感覚フィードバックであってよく、またユーザからの入力、音響、音声または触覚入力を含む任意の形式で受信されてよい。

30

【0105】

本発明の実施形態は、バックエンド構成要素、例えば、データサーバ等、を含む、またはミドルウェア構成要素、例えば、アプリケーションサーバ、を含む、またはフロントエンド構成要素、例えば、本発明の実装と相互作用できるようにするためのグラフィカルユーザインタフェースまたはウェブブラウザを有するクライアントコンピュータ、を含むコンピュータシステム中で実施することができ、もしくはそのようなバックエンド、ミドルウェア、またはフロントエンド構成要素の任意の組合せで実施することができる。システムの構成要素は、任意の形態または媒体の、通信ネットワーク等のデジタルデータ通信により相互接続することができる。通信ネットワークの例には、ローカルエリアネットワーク(「LAN」)およびインターネット等の広域ネットワーク(「WAN」)が含まれる。

40

【0106】

コンピュータシステムはクライアントおよびサーバを含むことができる。クライアントおよびサーバは、一般には相互に遠く離れており、通常は通信ネットワークを通して相互作用する。クライアントおよびサーバの関係は、それぞれのコンピュータ上で実行され、相互にクライアント-サーバ関係にあるコンピュータプログラムによって、発生する。

【0107】

本発明の特定の実施形態について説明をおこなった。その他の実施形態は添付の特許請求の範囲に含まれる。例えば、特許請求の範囲で列挙されるステップは、別の順序で実行

50

することもでき、それでも所望の結果を得ることができる。

【図面の簡単な説明】

【0108】

【図1】本発明の実施形態による、ユーザをクラスタリングする第1の方法を示すフローチャートである。

【図2】本発明の実施形態による、ユーザをクラスタリングする第2の方法を示すフローチャートである。

【図3】本発明の実施形態による、ユーザクラスタを使用する推奨者システムの動作を示すフローチャートである。

【図4】本発明の一実施形態による、ニュース推奨エンジンを有するニュースサービスを示す概略図である。

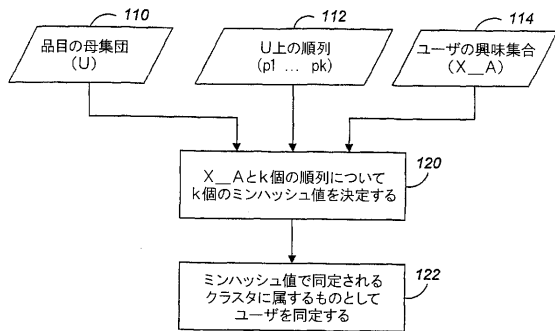
10

【符号の説明】

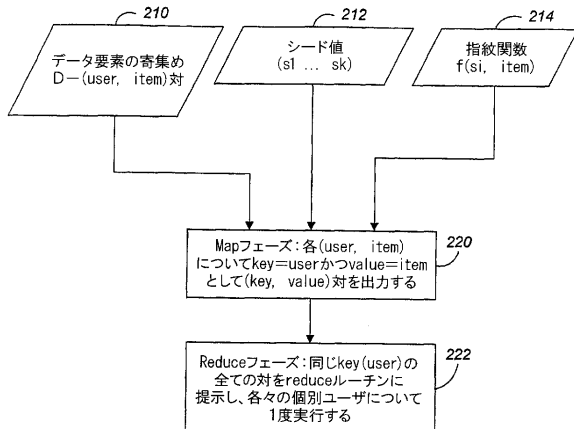
【0109】

- 402 a、402 b ユーザ
- 404 通信ネットワーク
- 410 ニュース推奨エンジン
- 420 ニュースサービス
- 430 ウェブサーバ
- 440 ログ

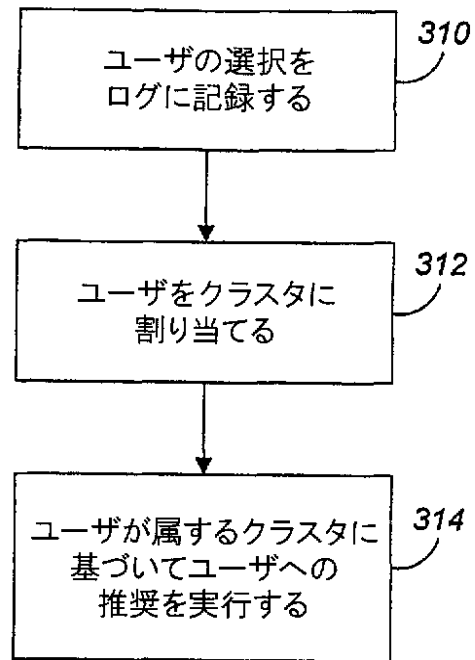
【図1】



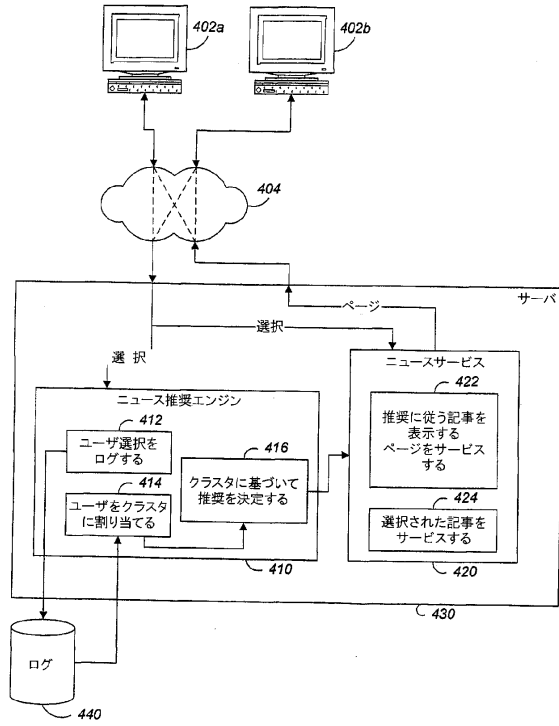
【図2】



【図3】



【 図 4 】



---

フロントページの続き

- (72)発明者 メイヤー・デイター  
アメリカ合衆国・カリフォルニア・95051・サンタ・クララ・サラトガ・アヴェニュー・13  
1・#3201
- (72)発明者 アシャトッシュ・ガーグ  
アメリカ合衆国・カリフォルニア・94086・サニーヴェイル・ファイヤーゾーン・テラス・8  
93

審査官 久々宇 篤志

- (56)参考文献 米国特許第06134532(US,A)  
米国特許第06697800(US,B1)  
Andrei Z.Broder 他, Min-Wise Independent Permutations, [online], 2000年 6月30  
日, pp.630-659, [平成24年7月6日検索]、インターネット, URL, <http://www.eecs.harvard.edu/~michaelm/postscripts/jcss2000.pdf>

- (58)調査した分野(Int.Cl., DB名)  
G06F 17/30