Vol.2020-IOT-48 No.12
2020/3/3

フォグコンピューティングによるグループ環境に適応したデータ管理基盤の提案

望月 龍一[1] 土屋 健[2] 広瀬 啓雄[2] 山田 哲靖[2]
澤野 弘明[3] 小柳 恵一[4]

概要：本研究は，フォグコンピューティングを用いた情報基盤により，テキストデータが分散している環境下において，ノード間の協調により独立して機械学習を用いたグループ特徴モデルの導出，その結合により効率的にデータを利用するためのネットワーク上での分析手法の同期を行う通信について検討を行っている．提案手法では，テキストデータの分散表現を対象とし，各ノードで生成，結合を行うために必要となる単語と ID の関連付けの同期，また各ノードで生成した分散表現の同期と結合を行う，二段階の同期によって特徴モデルの利用と最適化を行う．評価として，ノードにデータが分散した状況を想定したシミュレーションを行い，特徴モデルの結合により従来の集約的な解析と同程度の性能を有し，有効であることを示した．

キーワード：機械学習，分散情報プラットフォーム，特徴モデル結合，分析手法同期

Research on Data Management Platform based on Group Environments by Fog computing

MOCHIZUKI RYUICHI[1] TSUCHIYA TAKESHI[2] HIROSE HIROO[2]
YAMADA TETSUYASU[2] SAWANO HIROAKI[3] , and KOYANAGI KEIICHI[4]

1. はじめに

筆者らはこれまでフォグコンピューティング[1]によるデータ管理基盤について検討を行なってきた[2]．提案基盤は図 1 に示す形で，フォグノードと呼ばれるストレージと計算資源を持つノードを動的に構成する．この間で論理ネットワークを構築し，情報の共有を行う．この基盤の利用により，サービス利用者が生成するあらゆる情報は，サービス単位でクラウドに集約されるのではなく，個々のフォグノードにおいて分散し管理されることとなる．またそれら分散したデータの外部からの利用はフォグノードにより制限され，情報そのものが共有されるのではなく，フォグノードの計算資源により数値化，統計化されたデータのみが共有されることとなる．これにより，従来手法であるクラウドコンピューティングによるデータ集積と比較し，データの効率的な利用と，プライバシー性の維持を行う．一方で，現在行われている機械学習を用いた情報解析は，データを集約し同一の手法を適用することによって行われている．これと同じく，フォグコンピューティングによりデータがノード間に分散している前提の元，導出されたデータの利用を行うためには，各ノード間における解析手法の同期が必要不可欠であり，課題の一つとなっている．

本稿では，上述した情報基盤によりデータが分散している環境下において，ノード間の協調により機械学習を用いたグループ特徴モデルの導出，その結合により効率的にデータを利用するためのネットワーク同期手法を明らかにし，シミュレーションにより特性を明らかにする．

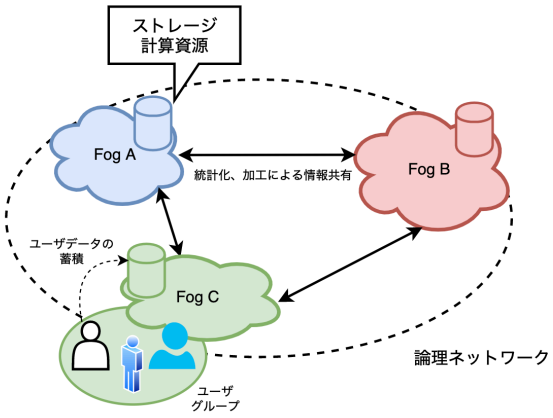

図 1 フォグコンピューティングによるデータ管理基盤

2. 関連研究

データを集約管理することなく分析を行うことを目的と

1 公立諏訪東京理科大学 経営情報学部
2 公立諏訪東京理科大学 工学部
3 愛知工業大学 情報科学部
4 早稲田大学 理工学術院

したプラットフォームとしてPDS(Personal Data Store)が提案されている[3]. PDS の元では，データがサービス単位でなくユーザ単位でクラウド上に分散して管理され，ユーザの許可を得ることによりサービスによる制限なくデータの取得と利用を行うことができる. これにより効率的なデータの利用と, アクセス制限によるプライバシ性の維持を行うことができる. 一方で，文献[3]の PDS プラットフォームにおいては，外部へデータの共有を行う際に数値化，統計化を行う処理機能を備えているが，データの管理，取得対象はユーザ単位に限られていることから，多数のユーザから取得したデータを元に分析を行う際には一度クラウド上に情報を集積する必要が生じる. そのため多数のユーザよりデータを取得，解析を行う際には従来の集約的な手法と同様の形態で処理を行う必要があるため課題となっている.

データがユーザやノードに分散している状態における機械学習を行うシステムである Federated Learning[3]が提案されている. この提案手法は図 2 に示すように，クラウド上に学習に用いるデータを集約し機械学習を行うのではなく，クラウド上にある特定のモデルをクライアントとなるノードに送信. 各ノードが所持するデータを用いて学習し，各自モデルを生成する. そして，クラウド上のモデルを更新する際は各ノードが導出したモデルを結合する. 具体的に, パラメータ調整に用いる勾配情報，すなわち変更点に相当するベクトル情報を，各ノードがクラウドに送信し，これを平均して統合することで最適化を行っている.

一方，この手法ではあるモデルの最適化をクラウド上でデータを集約することなく行うことを目的として，各ノードにデータを分散させ学習を行うものであるため，利用できる機械学習モデルはノード全体のモデルを平均した単一のものであり，クラウド上に集積済みのデータより生成したモデルがあることを前提としており，個々のノードにおいて分散したデータよりモデル構築を行うことについて検討されていない.

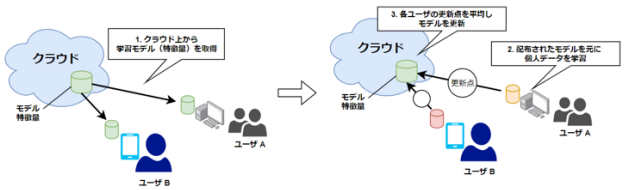

図 2　Federated Learning による機械学習の流れ

2.1 課題へのアプローチ

関連研究の動向を踏まえて，本提案手法では，フォグノード上で収集したテキスト情報を対象として，ノード間で分散し特徴モデルの生成とその最適化を行う手法について検討している.このとき，特徴モデルの生成を行うため 分散している文書データに含まれる単語の情報についてはノード間で統一させた後に特徴モデルの生成を行う必要がある. 従って,文書中の単語情報の同期処理，生成後の多数の特徴モデルの同期を行う必要がある.また，各ノードにおいて生成された特徴モデルの管理を行い結合する際，これをノード間で協調し執り行うプロトコルが要求される.

3. 特徴モデルの生成の最適化手法

本節では，グループ特徴モデルの導出のための情報同期プロトコルについて示す.

3.1 最適化対象の特徴モデル

本稿において最適化の対象とする特徴モデルは図 3 に示すように利用する単語ベクトルの語彙数$\boldsymbol{w}$，ベクトル次元数$\boldsymbol{h}$の行列モデル$\boldsymbol{W}$であり，各単語の分散表現となる. 単語ベクトル行はそれぞれv_0，v_1，… v_wの形で個々にIDを持ち，このIDと実際の単語文字列を関連付けてテーブル形式により持つ.これらをモデルデータとして各ノードが管理する文書情報をもとにして生成，管理，送受信を行うこととなる.

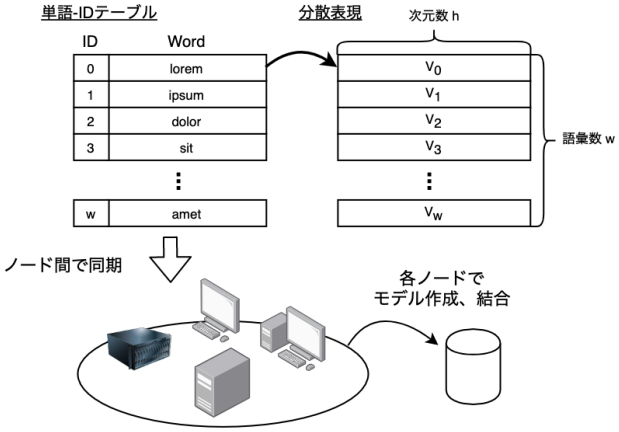

図 3　対象とする特徴モデル

各ノードにおいて独立して分散表現を生成し結合して利用する際には，同一の語彙数と次元数を持つ分散表現を各ノードにおいて生成する必要がある.そのため,各ノードにおいて分散して管理されている文書から，単語と ID の関連付けとベクトル次元数の情報を同期するための通信を行うことが要求される.

3.2 単語と ID の関連付け同期方法

単語と ID の関連付けの同期を行うためフォグノードを管理する管理ノードより，全てのノードの論理空間上のアドレスを取得し，該当するノード間でリング型のネットワークを構築，通信を行うことで情報同期を行う.

各ノードは，自ノード上で管理を行っている文書の形態素解析を行い，文書を単語の配列に分解した後，重複を取り除いたものを単語情報として次ノードに送信する.

各ノードにおいて単語情報は，図 4 の形でフォグノードに

おいて割り当てられた論理空間のアドレスに従い処理を行う．まず最もアドレスの小さいノードから，管理する情報に含まれる単語に ID を割り当て,リストを生成する.そして次のノードに送信する.次ノードは，前ノードまでにおいての単語情報に対して不足分の単語について順次 ID を割り当てる.これを末端のノードに到達するまで繰り返し行う.全てのノードの単語割り当てが完了したら，末端のノードは先端のノードに対して最終的な単語と ID の関連付け情報を送信したのち，末端のノードまで順次関連付け情報を送信する.分散表現のベクトル次元数$\boldsymbol{h}$についても，同期が完了した関連付け情報の送信を行うのと同時に,ノード全体に対して順次送信を行う.これによりすべてのノードで分散表現の生成に必要な情報が確定する.

各ノードにおいてプライバシ上の問題により公開が行えない単語が存在する場合は,単語情報を次のノードに送信しないことによって情報公開の制御を行う[2].これにより，分散表現そのもののプライバシ性を維持する.

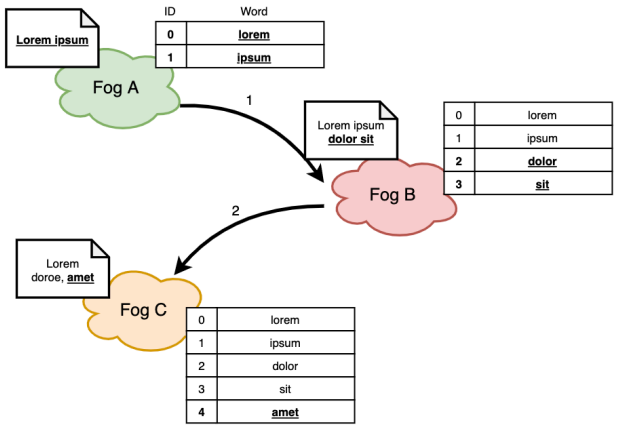

図 4　単語-ID の関連付け同期

4.　生成した特徴モデルの同期手法

本節では，各グループにて特徴モデルを導出，結合し最適化を行うための情報同期について示す.

4.1　分散表現の生成

個々のノードでの分散表現の生成は，テキストの特徴抽出に用いられる手法である Word2Vec[5]を用いる.Word2Vec では図 5 のように各ノードの文書情報を分かち書きし，単語と ID の関連付けを基に単語を ID に変換した数列データを入力とし，全単語について割り当てられた ID と対応する単語ベクトルを生成する.これを単語と ID の関連付け同期が完了した後に，各ノード独立して行い，それぞれが同一の単語 ID に対して異なる分散表現を獲得する.

学習済みの分散表現は，生成を行ったフォグノードのアドレス情報が含まれる一意な ID によって識別し，各ノードのストレージ上で管理する.ID には同期時の時系列情報を含み，同期を行った各ノードが生成した分散表現は統一して取得することができる.

分散表現を外部より取得する際は，管理ノードよりフォグノードのアドレス情報を取得し，各フォグノードに要求することによって行う.

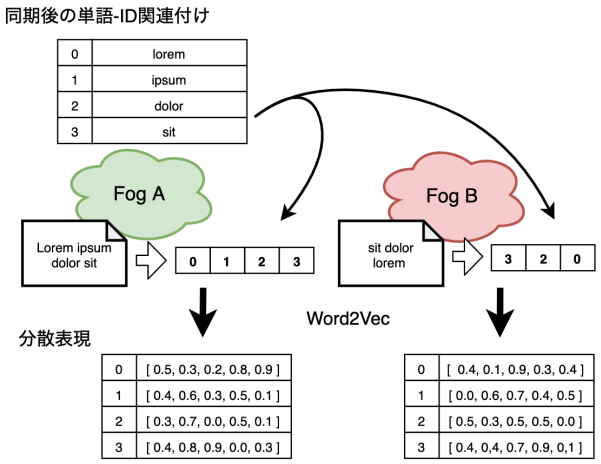

図 5　分散表現の生成

4.2　分散表現の結合

各ノードにおいて生成された分散表現を最適化のための事前学習モデルとして利用を行うためには，分散表現の同期，結合処理を行う必要がある.図 6 では簡単のため，二つのノードにおいて各単語 5 次元のベクトルで表される分散表現を結合する際の処理について示している.前節に述べた単語と ID の関連付けの同期によって，各ノードにおいて生成した分散表現の同じ ID と対応付けられるベクトルは，同じ ID と単語のものを表している.分散表現の結合においては，文献[3]における特徴量の結合手法を本提案手法でも採用し， 同一の単語ベクトルを一次元ごと平均することにより結合を行う.

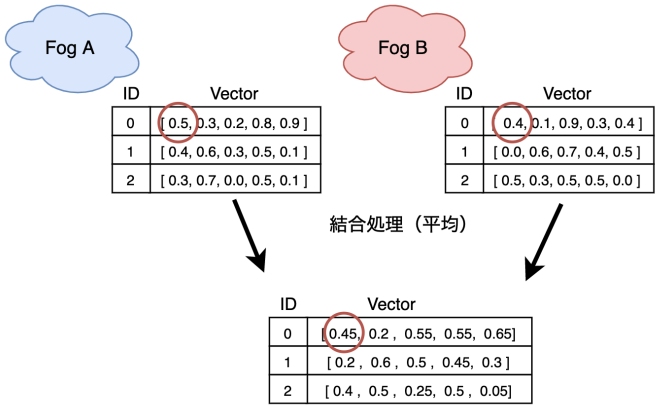

図 6　分散表現の結合処理

本稿は既存手法である集約型の大規模なグループのデータ処理，高スケーラビリティの要求される分析環境を置き換えることを目的としていることから，提案手法においても多数のノードの特徴モデルの結合処理を行うことが考えられ，コストを生じさせると考えられる.分散環境における効率的なデータ処理基盤は Apache Hadoop[6]や Apache Spark[7]などが実用化されており，本稿においては Apache Spark を採用し論理ネットワーク上のノードの計算資源を利用することにより分散表現の結合処理を行うことを想定している.

分散表現の結合は，論理ネットワーク上に存在する全てのノードが生成したものについて結合する場合，また選択的に結合する場合が考えられる.どのような基準に基づき分散表現の選択を行うかについては本稿では触れないが，論理ネットワーク上で結合対象とする分散表現を持つノードアドレス一覧をクエリとして,論理ネットワーク上に存在するいずれかのノードに対して要求を行い，この返答として結合済みの分散表現を取得する.

論理ネットワーク上で分散表現が行われたとき，要求を受け付けたフォグノードをマスターノード，プラットフォーム上の他のフォグノードをスレーブノードとして，クラスタ環境を構築する.マスターノードは分散表現の要求先であるアドレスに基づいて順次各ノードに分散表現の要求を行い,分散表現を取得する．その後，クラスター上で各グループの分散表現を単語ごと集計(Reduce)し，平均(Map)を行うジョブを生成し,その結果として得られる結合した分散表現を利用する.

結合済みの分散表現は，各フォグノード内において提供するサービス，分析の利用目的に応じてノード内のデータを用いて追加学習を行うことによって最適化を実行し利用する.

5. 特徴モデル結合の評価

結合したモデルの性能を従来の集約的なモデルとシミュレーション比較し，有効性を明らかにする.

本評価では，表 1 の環境で提案手法を実装している.論理ネットワーク上にデータが分散した状況を想定して，英 Wikipedia のダンプファイル[8]を用いる．このとき,記事をランダムに各論理ネットワーク上に配置し,フォグノードで Word2Vec を用いて特徴モデルの生成を行う.この特徴モデルをランダムに結合する.

結合した特徴モデルを類似文書検索の最適化に評価に適用した.このとき,ニュース記事とそのトピック分類カテゴリがラベリングされたデータセットである newsgroup[9]のうち，5 ラベルのトピック分類を行うタスクを実施し，精度の比較を行う.文書分類に用いる文書特徴量は，文書中に出現する各単語を分散表現より参照し，その平均値とした.

本稿では，図 7 に示す形で、

1) Wikipedia 文書データより 100000 件をランダムに選択し，$N(1 < N \leq 20)$ノードに分散配置して，従来手法($N = 1$)の場合とモデルを結合した場合においての精度の比較
2) 20 ノードに Wikipedia 文書を各 10000 件ランダムに配置し学習させ，生成した特徴モデルを 1 ノード数ずつ増加させながら結合した場合においての精度の比較と，モデル結合時の処理時間

の二点について評価を行った.

表 1 評価環境

実装環境	Arch Linux Python 3.7.4
分散表現の生成	Gensim Word2vec[9] (dim=300,windowsize=5, epoch=5)
文書分類処理	LightGBM
事前学習データ	Wikimedia Database backup dumps (enwiki-20200101)[8]
文書分類対象データ	newsgroups[10]

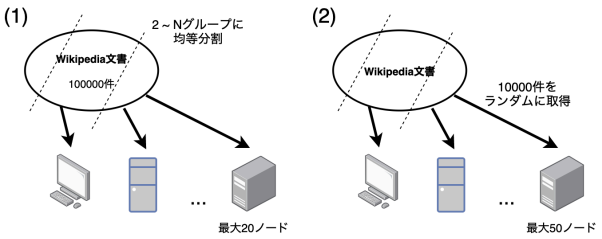

図 7 シミュレーション環境

5.1 評価結果

シミュレーション評価の結果を図 8 および図 9 に示す.どちらの図も，x軸は結合したモデル数であり，y軸に文書分類の精度を示している.

図 6 は Wikipedia 文書データより 100000 件をランダムに選択し，$N(1 < N \leq 20)$ノードに対して均等に分散配置した際の精度を示している.$N = 0$の時はノード間での分散表現の同期を行わずに文書分類を行った際の精度であり，$N = 1$は従来手法である，最適化に用いる分散表現を生成するためのデータをすべて集約して単一のモデルを作成した場合の精度である.はじめ，$N = 2,\ 4,\ 5$程度の少ない分割数の規模では学習が不安定となり従来手法，また分散表現の同期を行わない場合と比較しても精度が低下する傾向が見られるが，$N = 10,\ 16$つまり各ノード 10000 記事程度に分割数を大きくした場合，精度が従来手法と同程度かそれ以上となる傾向が見られる.

精度が低下する部分については, Wikipedia の記事データよ

りランダムに選択した乱雑性，一般性を持つデータであることから，学習を行った際のデータの偏りが考えられ，また現在の Word2Vec による学習では，各次元において潜在的に表現されている特徴は異なる.これが分割数が極端に少ない場合においては平均化によって特性が顕著に失われてしまったことが要因として考えられる.

いずれの場合においても，性能への影響範囲は$\frac{1}{100}$を超えないことから，誤差の範囲であるとみなすことができる.本提案手法は，従来手法のデータ集積による分散表現の計算と対照的に， 各ノードでのデータ学習後， 結合の処理と最適化のための追加学習が行われている. この結合処理と追加学習の点において計算とノード間通信コストが増大することになるが， データ集約に伴うプライバシ性を解決しながら， 従来手法と同等の性能の分析を行うことができる利点を持つと考えられる. 今後， 提案手法をより規模の多いネットワーク， 分析環境に適用した場合のコストの点から定量的に評価を行う予定である.

図 9 は 50 ノードに Wikipedia 文書を各 10000 件ランダムに配置し学習を行い，結合数に対する精度を示したグラフである.直線は比較のため，従来手法である単純に集約を行った特徴モデル結合なし($N = 0$)の精度を表している.モデル結合数と精度に相関は見られず揺らぎがあるが，全体として従来手法より精度が下回る傾向は少ない.結合数によっては，精度が 1%以上向上する場合も確認でき，最適な特徴モデルを選択することによる更なる最適化の可能性も示唆できる.図 10 は 50 ノードまでのモデル結合時にかかる処理時間のグラフであり，x軸は結合したモデル数.y軸に処理秒数を示している.処理時間は線形の傾向を示しており，一ノードを追加するにあたり約 8 秒ごと増加する.この結果は図 9 において生成した分散表現の各ノードあたり3×10^6個の語彙数の単語ベクトルを結合したものであり，ノード数の増加に伴って語彙数，分散表現の増加によってさらに負荷が増大することが懸念される.これについて，各フォグノードにおいて最適なモデルのみを選択することにより，モデル結合に要する計算コストの削減が期待でき，また最適化が可能な特徴に適合する形で適切なノード，論理空間上においてモデルの管理を行うことにより，通信コストの削減を図ることができる，この点においてモデル結合の特性について今後検討を行う予定である.

以上のことから，提案手法はデータが分散している環境下においても，特徴モデルを生成，結合することを通して既存手法と変わらない精度で分析の最適化が行えることが分かる.

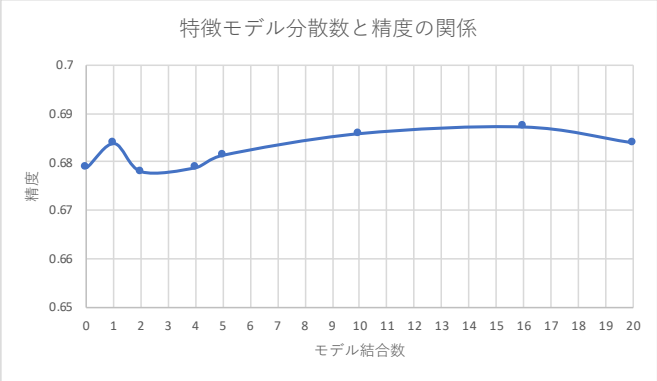

図 8　特徴モデル分散数と精度の関係

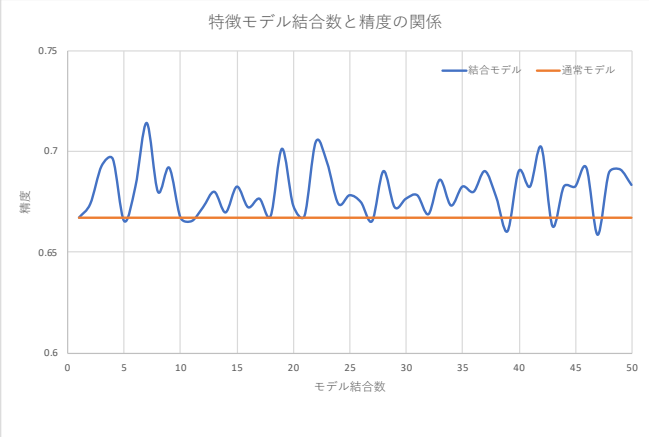

図 9　特徴モデル結合数と精度の関係

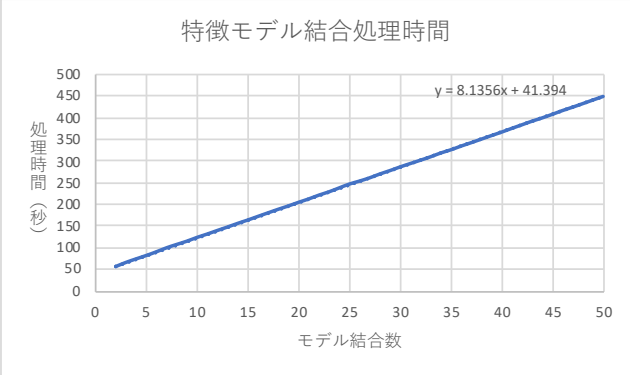

図 10 特徴モデル結合処理時間

6. 今後の課題

評価により，特徴モデルの結合により従来の集約的な解析と同程度の性能を有し，有効であることを示したが，実際に稼働するサービスにおいては，グループ間のどのような特徴モデルを結合することにより最適化が行えるのか明らかでない.また適切なモデルの選択が可能となることで結合にかかるコストを抑制できることから，分散した特徴モデルを結合するという環境を前提とした，特徴量の特性について早急に分析を行い，定量的な評価を行う必要がある点

が課題である.また特徴量は時系列的に同列のものが結合, 利用されるとは限らないため, 生成した特徴量をノード間で効率的に管理する手法, 具体的に特徴量の類似性に基づいて論理空間上で管理を行う手法が求められる.これについても検討を行なう.

また, 図 8 における評価では, 既存の Word2Vec によって個別のノードで学習を行った場合では, 各次元において潜在的に表現されている特徴は異なるため, 分割数が極端に少ない場合においては平均化によって特性が顕著に失われて精度が低下している可能性について明らかになった.すなわち, ノード間において単語を示す ID 情報の同期だけでなく, 各ベクトルにおける表現の要素についても同期を行うことが可能となれば, より効率的な特徴モデルの利用が行えると考えられる.このような特徴量に含まれる潜在空間を特定する手法として特徴表現の解きほぐしが提案されており[11], これを拡張することによって同期を行う手法について考えている.

7. まとめ

本稿では, フォグコンピューティングを用いた情報基盤により, テキストデータが分散している環境下において, ノード間の協調により機械学習を用いたグループ特徴モデルの導出, その結合により効率的にデータを利用するためのネットワーク上での同期手法について示した. 提案手法ではテキストデータ用いた特徴モデルを対象として, 分散表現を独立したノードで生成, 結合を行う手法について明らかにした. 具体的に, 分散表現の生成にあたって必要となる単語と ID の関連付け, ベクトル次元数の情報を同期し, 各ノードで独立して分散表現を生成する. そして各ノードで生成した分散表現を同期, 結合を行う形の, 二段階の同期によってデータ分析の最適化を行う.

論理ネットワーク上に文書データが分散している環境を想定したシミュレーション評価により, 特徴モデルの結合により従来の集約的な解析と同程度の性能を有し, 有効であることを示した.

課題として, 特徴モデルの結合数と最適化の精度は依存しない.このためグループ間においてどの特徴モデルを結合することにより最適化が行えるのか明らかにすることにより, 特徴モデルの結合にかかるコスト,また効率的なモデル管理が行えることが期待できる. 今後, 分散した特徴モデルを結合するという環境を前提とした, 特徴量の特性について分析を行った後, 情報基盤としての実装, 実際のデータを用いた分析を行う.

参考文献

1) F. Bonomi, R. Milito, P. Natarajan, J. Zhu: "Fog computing: A platform for Internet of Things and analytics" Big Data and Internet of Things: A Roadmap for Smart Environments, Cham, Switzerland:Springer, pp. 169-186, Mar.2014.
2) 土屋，望月，広瀬，山田，澤野，小柳，モデル結合による分散データ環境で機械学習を可能とする分散データプラットフォームの検討，信学技報 LOIS,2020.3 (印刷中)
3) Yves-Alexandre de Montjoye ,Erez Shmueli, Samuel S. Wang, Alex Sandy Pentland, “openPDS: Protecting the Privacy of Metadata through SafeAnswers”,https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0098790, (checked on Jan. 20th. 2020)
4) Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, Dave Bacon, “Federated Learning: Strategies for Improving Communication Efficiency”, https://arxiv.org/abs/1610.05492, (checked on Jan. 20th. 2020)
5) Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space", CoRR, Vol. abs/1301.3781, 2013.
6) Apache Hadoop, “https://hadoop.apache.org/”
7) Apache Spark, “https://spark.apache.org/”
8) Wikimedia Downloads (Database backup dumps), https://dumps.wikimedia.org/backup-index.html (checked on Jan. 1th. 2020)
9) Gensim, “https://radimrehurek.com/gensim/”,
10) 20newsgroups, “http://qwone.com/ jason/20Newsgroups/” (checked on Jan.18th. 2020)
11) Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, Olivier Bachem, Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, https://arxiv.org/abs/1811.12359, (checked on Jan. 20th. 2020)