社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

群知能を用いたクラスタリング手法のトラヒック分類への適用と評価

須恵　匠†　大下　裕一†　村田　正幸†

† 大阪大学 大学院情報科学研究科　〒 565–0871 大阪府吹田市山田丘 1–5
E-mail: †t-sue@ist.osaka-u.ac.jp

あらまし　性能要求に応じてトラヒックの分類・識別を行う手法のとして，トラヒックの統計情報をもとにした手法が注目されている．これらの手法では各フローの統計情報を取得し，機械学習を用いて各フローの分類を行う．機械学習手法の中でも，特にクラスタリングと呼ばれる類似のデータを集めたクラスタを構成する手法は事前に各分類先の特徴を学習することなく分類を行うことが可能である．クラスタリング手法に関する研究の中でも特に生物学に知見を得た手法が，クラスタリングの正確性と計算量のバランスの良さから注目を集めている．これらの新たなクラスタリング手法をトラヒック分類に適用することにより，正確な分類と短い計算時間での分類の両立が可能となると考えられる．しかしながら，これらの新たな手法をトラヒック分類へ適用した際の性能については，十分な議論が行われていない．本稿では，Web アプリケーションによるトラヒックの分類に焦点をあて，トラヒック分類機に生物学の知見にもとづいた分類手法の適用性について評価を行う．評価の結果，生物学の知見に基づいた AntTree を用いることにより，次々と到着するフローをアプリケーションの種類ごとに分類することが可能であり，全フローの情報を取得後に K means++法で分類した場合と同程度の分類精度を達成することができることが明らかになった．
キーワード　トラヒック分類，クラスタリング，群知能，Web アプリケーション

Application and Evaluation of Clustering Methods using Swarm Intelligence to Traffic Classification

Takui SUE†, Yuichi OHSITA†, and Masayuki MURATA†

† Graduate School of Information Science and Technology, Osaka University　Yamadaoka 1–5, Suita-shi, Osaka, 565–0871 Japan
E-mail: †t-sue@ist.osaka-u.ac.jp

Abstract　One approach to classify the Internet traffic is to use the statistical information of traffic. In this approach, the traffic is classified based on the statistical information by the machine learning. One of the popular machine learning techniques is clustering. In recent years, many clustering methods have been proposed. Among them, the bio-inspired clustering methods can classify the data accurately with a small computational complexity/ However, the applicability of such new clustering methods to the traffic classification is not sufficiently discussed. In this paper, we evaluate one of the bio-inspired clustering methods called AntTree when it is applied to the traffic classification. The results show that the AntTree can classify Web application traffic accurately even when it is applied to the online classification.
Key words　Traffic classification, Clustering, Swarm Intelligence, Web application

1. はじめに

インターネットの普及に伴い多種多様なサービスがインターネットを介して提供されるようになり，サービスのネットワークに対する性能要求も多岐にわたるようになってきた．

ネットワーク管理者はアプリケーションによって異なる性能要求を満たしつつ，各アプリケーションのトラヒックを収容する必要がある．そのため，異なる性能要求を持つサービスを一つのネットワーク上に収容する手法の検討が進められており，たとえば文献 [1] では，サービスごとに仮想ネットワークを構築し，各サービスの要求にあわせて各仮想ネットワークを動的に制御しつつ仮想ネットワーク間の資源を調停する手法が提案されている．このようなアプリケーションの性能要求に合わせたネットワーク制御を行うためには，ネットワーク内を流れる

トラヒックをアプリケーションに合わせて分類することが必要となる．

従来，トラヒックのアプリケーション識別は， TCP・UDP のポート番号をもとに行われてきた [2]．しかしながら近年，YouTube などの動画共有サービスからゲームなどのインタラクティブなアプリケーションまで，多種多様なサービスが Web ブラウザを介して提供されるようになってきた．その結果，ネットワークへの性能要求の異なる多種多様なサービスが HTTP プロトコルを用いて提供されるようになってきており [3]，それらはすべて 80 番ポートや 443 番ポートを用いて通信を行うため，ポート番号によるトラヒックの種別の分類は困難となってきている．

ポート番号によらずトラヒック識別を行う手法のとして，トラヒックの統計情報をもとにした手法が注目されている [2], [4]〜[6]．パケットサイズやパケットの到着間隔の平均や分散，分布はアプリケーションや通信に用いられるプロトコルによって大きく異なることが知られている．統計情報を用いた手法ではこれを利用し，各フローについてパケットのサイズや到着間隔などの統計情報を取得し，その統計情報をもとにフローの分類を行う．統計情報取得後，各フローの分類には機械学習が用いられる．特に，クラスタリングと呼ばれる類似のデータを集めたクラスタ (塊) を構成する手法は，事前に各分類先の特徴を学習することなく分類を行うことが可能であり広く用いられている [4], [6]．統計情報を用いたフローの分類では分類に用いる機械学習手法が大きな影響を与え，同じ統計データを用いた場合であっても用いた機械学習手法によって精度が異なり，また，分類にかかる時間も機械学習手法によって異なる．トラヒック分類に用いる機械学習手法に関する検討も行われているものの [6]，通信に用いられているアプリケーションプロトコルの識別に焦点をあてた手法の検討がほとんどであり，Web アプリケーションに関するトラヒックについて，そのネットワークへの性能要求に応じた分類を行うことを目的とした分類に焦点をあてた検討は十分に行われていない．

さらに近年，データマイニングの分野においてクラスタリング手法に関する研究も進められており，中でも生物学に知見を得た手法は計算量とクラスタリングの正確性のバランスの良さから注目を集めている [7]．これらの新たなクラスタリング手法をトラヒック分類に適用することにより，分類の正確性と短い計算時間での分類の両立が可能となると考えられる．しかしながら，これらの新たな手法をトラヒック分類へ適用した際の性能について十分な議論は行われていない．

本稿では， Web アプリケーションによるトラヒックの分類に焦点を当て，トラヒック分類機に生物学の知見にもとづいた分類手法の適用性について評価を行う．特に，トラヒックの分類では，フロー情報が次々に到着する環境下において，適切な分類を行うことが求められる．そこで，本稿では，生物学の知見に基づいた分類手法を，フロー情報が順に到着する環境において評価を行った．本評価では評価にあたり， Web アプリケーションを利用している際に発生したパケットをキャプチャし，キャプチャしたパケットトレースデータから分類に用いるトラヒック統計情報を取得した．その後，取得したトラヒック統計情報を順に与え，クラスタリングを行い，その正確度を評価した．

以降，2. でクラスタリング手法について紹介する．その後，本稿で用いたクラスタリング手法を適用したトラヒック分類手法について 3. で述べ，4. において，クラスタリング手法の評価を行う．最後に 5. で，まとめと今後の課題について述べる．

2. クラスタリング

2.1 クラスタリングの概要

クラスタリングはデータの集合を入力とし，入力されたデータを類似するデータを集めた複数のクラスタに分割する手法である．各データは N 個の値を持ち (N 次元) ，N 個の値を用いてデータ間の類似度が定義される．そしてクラスタリング手法では，それぞれのクラスタに類似度の高いデータが集まるようにクラスタが形成される．

様々なデータ間の類似度の定義が存在するが，その中でもよく使われるのがユークリッド距離である．データ x とデータ y のユークリッド距離は以下で定義される．

$$Dist(x,y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{1}$$

ただし，x_i はデータ x の i 番目の値を示す．以降の本稿においてもクラスタリングを行う際には，類似度の定義としてユークリッド距離を用いる．

2.2 従来型クラスタリング手法：K-means 法

クラスタリング手法で最も広く使われている手法の一つが K-means 法である．K-means 法はクラスタ数 k を入力パラメータとし，以下の手順により入力されたデータ集合を k 個のクラスタに分割する．

（1） 各データを k 個のクラスタにランダムに分割する

（2） 割り振られたデータをもとにクラスタ内のデータの算術平均を計算し，得られた値をクラスタの中心と定義する

（3） 各データについて，各クラスタの中心との距離を計算し，当該データをもっとも中心と距離が近いクラスタに移動する

（4） データの移動が発生しなかった場合は処理を終了，データの移動が発生した場合は手順 2 に戻る

K-means 法の結果は，手順 1 で行う各データのランダムなクラスタへの割り当てに依存する．そのため，よりよい結果を得るために初期値の与え方を変えた K-means++法 [8] が提案されている．

K means++法は，上記の K-means 法の手順 1 の変わりに以下の手順を行い，初期のクラスタの中心を計算する．

（1） データからランダムに 1 つ選びそれをクラスタの中心とする

（2） それぞれのデータ x 対して，各クラスタの中心との距離を計算し，その距離の最小値を D_x とする

（3） 各データ点 x に対して，D_x^2 に比例する重み付きの確率分布を用いて，新たなデータを選択し，選択したデータをク

ラスタの中心とする

（4） 手順 2，3 を繰り返し，k 個のクラスタの中心を選択する．

k 個のクラスタの中心が選択された後は，通常の K-means 法の手順によりクラスタリングを行う．

本稿では，より正確なクラスタリングが可能な K-means++ 法を評価対象のクラスタリング手法の一つとして用いる．

2.3 群知能に基づくクラスタリング手法：AntTree

群知能は動物の群れの行動に着想を得た計算手法であり，近年多くの手法が研究されている．クラスタリングの分野でも群知能を用いた手法の検討が進められている．群知能に基づく代表的なクラスタリング手法がアリの行動を模倣した AntTree [7], [9] である．この手法はアリが互いに寄り集まって鎖状につながり，複雑な構造を形成する行動をもとにしたものである．アリはスタート地点 (サポート) から出発する．一部のアリがサポートとつながり始め，つながったアリにさらに別のアリが繋がる．繋がったアリは組織の一部となり，他のアリはその組織を伝って移動するようになり，さらに別のアリとつながる．この手順が繰り返されることにより，複雑で大規模な組織が構築される．

AntTree ではこのアリの行動様式を模して各データをアリとみなし，アリの行動様式に従ってデータが他のデータとつながる．本手法では，データ同士がつながる際に類似のデータのみがつながるようにすることにより，類似のデータがつながった鎖を構築する．

各アリ (データ) は，二つの閾値 T_S と T_D を持つ．また，各アリは他の一匹のアリのみと繋がることができ，現在鎖状に組織を構成したアリの上を移動しつつ，どのアリと繋がるのかを以下のルールに従って自律的に判断する．

- 現在地がサポートである場合
  - サポートと繋がったアリがいない場合はサポートと繋がる
  - サポートと繋がったアリがいる場合は
    - サポートにつながったアリのうち，もっとも自身と似たアリを探す
    - もっとも自身と似たアリとの類似度が T_S よりも大きければ，そのアリに移動する
    - もっとも自身と似たアリとの類似度が T_D よりも小さければ，サポートにつながる
    - それ以外は，$T_S \leftarrow \alpha_1 T_S$，$T_D \leftarrow T_D + \alpha_2$ として閾値を更新する
- 現在地がサポート以外であれば，
  - 現在地のアリとの類似度が T_S よりも大きければ
    - 現在地のアリと繋がっているアリのうち，もっとも自身と似たアリを探す
    - もっとも自身と似たアリの類似度が T_D よりも小さければ，現在地のアリと繋がる
    - もっとも自身と似たアリの類似度が T_D よりも大きければ，$T_S \leftarrow \alpha_1 T_S$，$T_D \leftarrow T_D + \alpha_2$ として閾値を更新した上で，ランダムに選択した現在地のアリと繋がっているアリに移動する
  - 現在地のアリとの類似度が T_S 以下であれば，ランダムに選択した現在地のありと繋がっているアリに移動する

上記のアルゴリズムにおいて T_S，T_D の初期値，α_1，α_2 はクラスタの形成のされやすさを決めるパラメータである．以後の本稿の評価では，T_S の初期値は 1，T_D の初期値は 0 とし，α_1 は [実際の評価に用いた値]，α_2 は [実際の評価に用いた値] とした．

また，AntTree では類似度は 0 から 1 の間で表現される値である．xy 間の類似度 $Sim(x,y)$ を $Dist(x,y)$ と $Dist_{max}$ を用いて，以下のように定義することにより，0 から 1 の間の値をとる類似度を定義することができる．

$$Sim(x,y) = 1 - \frac{Dist(x,y)}{Dist_{max}} \tag{2}$$

$$Dist_{max} = max(Dist(\forall x, \forall y)) \tag{3}$$

そして，この手順により，全アリがいずれかのアリにつながった時点でクラスタリングを終了する．その時点で，サポートにつながっているアリの部分木をそれぞれのクラスタとみなす．

AntTree では，各データが自律的に自身の接続先となるアリを探索し，接続することによりクラスタリングが行われる．そのため，クラスタリング対象の全データが揃っていない状況においても，現在揃っているデータに対応するアリのみが接続先を探索し接続をする，新たなデータが到着する度に，そのデータに対応するアリを投入する形で，クラスタリングを行うことができる．そのため，随時トラヒックデータが到着し，そのトラヒックデータのクラスタリング・分類が必要な，ネットワークトラヒックのオンライン型の分類に適していると考えられる．そこで，本稿では，AntTree のオンライン型トラヒック分類への適用について議論を行う．

3. トラヒック分類への適用

3.1 概　　要

本稿でのトラヒックの分類は分類器において行われるものとする．分類器はネットワークのアクセスルータに設置され，アクセスルータを流れるトラヒックを観測する．観測は，同一送信元 IP アドレス・宛先 IP アドレス間の一連の通信をフローとして定義し，フロー単位で行う．フローは当該 IP アドレス間の最初のパケットが到達した時点で開始したとみなされ，一定期間当該 IP アドレス間の通信が行われなかった場合にフローは終了したとみなされる．各フローに対する統計情報はフロー終了検出時に集計され，クラスタリング手法の入力として使われる．そしてクラスタリング手法を用いて各フローの分類を行う．

クラスタリング手法を適用することにより，各フローは類似したフローを集めたクラスタに分けられる．その後，クラスタ内のフローが関係するアプリケーションにもとづき各クラスタにラベルを付けることにより，各フローの分類を完了する．

3.2 分類に用いる統計情報

本稿では， Web アプリケーショントラヒックの分類を目標としている．そこで本節では， Web アプリケーションで発生するトラヒックの性質について説明し，その後分類に用いる統

計情報について説明する．

Web アプリケーションによる通信は，すべて HTTP を用いて行われる．HTTP は次の手順により通信が行われる．まず，クライアントからサーバに対して TCP の接続要求を送り接続を確立する．接続を確立した後は，クライアントからデータの取得を要求する GET や，データの送信を行う POST リクエストを行う．サーバ側はクライアントからの要求に応じて，レスポンスとしてクライアントが要求したデータを返答する．

Web アプリケーションによる通信は，アプリケーションの種類によらず上記の手順に従う．しかしながら，アプリケーションによってクライアントが GET や POST のリクエストを行うタイミングや，各リクエストに対するレスポンスの大きさが異なる．そこで本稿では，そのような特徴を捉えることができるようなトラヒック統計情報を用いてアプリケーションの識別を行う．

本稿では， Web アプリケーションの挙動に起因する以下の指標を用いてクラスタリングを行う．

上下方向のペイロード付パケット数の比　本稿では， Well-known Port 宛の通信を上り， Well-known Port からの通信を下りと定義する．そして，上り方向のペイロード付パケット数 P^{UP} と，下り方向のペイロード付パケット数 P^{DOWN} をカウントし，以下の式で定義される値を計算する．

$$\log \frac{P^{DOWN}}{P^{UP}} \tag{4}$$

ペイロード付パケットのみをカウントすることにより， ACK のみのパケットは除外され，上り方向のクライアントからのリクエスト関係するパケット数，下り方向のサーバらからのレスポンスに該当するパケット数のみを数えることができる．そのため，式 (4) の値は，クライアントからのリクエストとサーバからのレスポンスに該当するパケットの比率となる．この値が大きければ，少ないクライアントのリクエストに対して多量のレスポンスが返されているとみなすことができる．その一方，この値が小さければ，クライアントは多くのリクエストをサーバに対して送信しており，インタラクティブな通信や，クライアントからのデータのアップロードが行われているとみなすことができる．

下り方向のペイロード付パケット到着レートの変動係数　同一サーバから同一クライアントへ送られているトラヒックのうち，ペイロード付パケットの到着レートを計測する．本稿では，パケットの到着レートは各タイムスロットあたりに到着したパケットの数と定義し，4. 章の評価の際には，タイムスロットの長さを 5 秒とした．R_i^{DOWN} を i 番目の下り方向のペイロード付パケットの到着レートとし，該当フローが通信を行っている時間が M 個のタイムスロットに分割できるとすると，当該フローの平均パケット到着レートは以下のように定義される．

$$Avg^{DOWN} = \frac{1}{M}\sum_{i=0}^{M} R_i^{DOWN} \tag{5}$$

また，当該フローの下り方向ペイロード付パケットの到着レートの標準偏差は以下のように定義される．

$$Std^{DOWN} = \sqrt{\frac{1}{M-1}\sum_{i=0}^{M}(R_i^{DOWN} - Avg^{DOWN})} \tag{6}$$

変動係数は，Avg^{DOWN} と Std^{DOWN} を用いて以下のように定義される．

$$CV^{DOWN} = \frac{Std^{DOWN}}{Avg^{DOWN}} \tag{7}$$

CV^{DOWN} はサーバからのレスポンスパケットの到着レートのばらつきを表す．CV^{DOWN} が小さければ，一定のレートでレスポンスパケットが到着し続けていることを示し，ストリーミング系のアプリケーションなど，下り方向で一定のレートでの通信を行い続けるフローであると判断することができる．

上り方向のペイロード付パケット到着レートの変動係数　同一クライアントから同一サーバへ送られているトラヒックのうち，ペイロード付パケットの到着レートを計測する．そして下り方向のペイロード付パケットの到着レートと同様，その到着レートの平均，標準偏差をもとに変動係数を以下のように計算する．

$$Avg^{UP} = \frac{1}{M}\sum_{i=0}^{M} R_i^{UP} \tag{8}$$

$$Std^{UP} = \sqrt{\frac{1}{M-1}\sum_{i=0}^{M}(R_i^{UP} - Avg^{UP})} \tag{9}$$

$$CV^{UP} = \frac{Std^{UP}}{Avg^{UP}} \tag{10}$$

CV^{UP} はクライアントからのリクエストパケットの到着レートのばらつきを表す．CV^{UP} が小さければ，一定のレートでクライアントからリクエストが送られていることを示す．ストリーミング系のアプリケーションでは，ストリーミングのデータは小さなデータに分割され，クライアントはサーバに対して次のデータに対するリクエストを定期的に送信する．CV^{UP} を用いることにより，そのような定期的にサーバに対してリクエストを送っているようなフローを検出することが可能となる．

4. 評　　価

4.1 分類対象のトラヒックデータ

本稿では，大阪大学内の PC からブラウザ (Internet Explorer 11) を用いて Web アプリケーションを利用し，その際に発生したパケットを当該 PC 上で動作させたネットワークアナライザソフトウェア Wireshark [10] を用いてキャプチャしたデータを用いる．

本稿では，ネットワークへの要求の異なる以下の 5 種類の Web アプリケーションを利用した際のパケットキャプチャデータを用いた．パケットキャプチャは，各種類の Web アプリケーションあたり 20 回行った．

Download　ユーザが所望のファイルをダウンロードする Web アプリケーションであり，クライアント側からのリクエストに対して，サーバはリクエストに指定されたファイルの送信を行う．当該アプリケーションでは，ユーザが高速に必要なファイ

ルを取得することが望まれるため，サーバからクライアントへの十分に大きな通信帯域が確保されることが望ましい．一方，上り方向はファイル取得のリクエストしか送信されないため，大きな帯域の確保は不要である．本稿では，Bitcasa [11]，Box [12]， Dropbox [13] の 3 種類のクラウドサービスからファイルをダウンロードした場合のパケットキャプチャデータを用いた．

Upload ユーザがクラウドサービス等にファイルをアップロードし，別の端末や他のユーザとの共有を行うアプリケーションである．ファイルのアップロードを行う際には，クライアントからサーバに多量のトラヒックが発生する一方，サーバからクライアントへ送られるトラヒック量は少ない．そのため，クライアントからサーバ方向の通信に対して十分に大きい帯域を確保することが望まれる．本稿では，Download と同じく，Bitcasa [11]， Box [12]， Dropbox [13] の 3 種類のクラウドサービスに対してファイルをアップロードした際にキャプチャを行ったデータを用いた．

Video Live Streaming(Live) イベントを撮影した動画をリアルタイムで配信するサービスである．クライアント側は定期的に次の時刻の動画に対するデータを要求するリクエストをサーバ側に送信し，サーバはクライアントの要求に合わせてクライアントに新たに撮影された動画を送信するという形で，Web アプリケーションとして動画のストリーミングサービスが実現されている．本アプリケーションでは，各時刻で撮影された動画のデータがサーバからクライアントに送信され，クライアントからサーバへは新たな時刻の動画データの要求が定期的に送信される．本サービスは，ネットワークに対しては下り方向で各時刻の動画データを送信するだけの帯域が安定的に提供されることを要求する．本稿では，Ustream [14] の生放送を視聴した際に発生したパケットをキャプチャしたデータを用いる．

Video on Demand(Video) ユーザが所望した動画ファイルをダウンロードして再生を行うアプリケーションである．本アプリケーションも，Video Live Streaming と同様ひとつの動画は複数のデータに分割され，その分割された各データに対してクライアントが順にリクエストを送ることにより必要な動画像データを受信しながら再生を行うという形で，Web アプリケーションとして実現されている．Video on Demand は，Video Live Streaming とは異なり現在撮影された動画像ではなく，サーバ側に蓄えられている動画データが送信される．クライアント側では，サーバから送られてきた動画像データをキャッシュしながら再生を行う．そのため，十分な帯域が存在していれば送信可能な分の動画像データがクライアントに送られる．本稿では，YouTube [15]，Ustream [14] の録画放送配信，ニコニコ動画 [16] を利用した際のパケットをキャプチャしたデータを用いた．

Interactive ユーザの操作に応じてサーバ側にリクエストが送られ，ユーザの操作に応じた返答がサーバから行われるようなアプリケーションである．本稿では，このようなアプリケーションとして地図情報サービスを用いた．地図情報サービスでは，ユーザがクライアント端末側で表示された地図を動かした際には，動かした先の新しい地図情報をサーバ側から取得して表示を行う．このアプリケーションではユーザが操作したタイミングでサーバ側にリクエストが送られ，リクエストされた情報をサーバが返答する．このようなユーザの操作に応じてサービスを提供するアプリケーションでは即応性が重要であり，低遅延の通信が必要となる．本稿では，Google マップ [17]，Yahoo!地図 [18] を利用した際にキャプチャしたデータを用いた．

4.2 評価指標

Web アプリケーションの分類結果がネットワーク制御の入力として用いることを考えると，各フローが自身のアプリケーションとはネットワークに対する性能要求が異なるフローと同一のクラスタに分類されることは避ける必要がある．そこで，本評価では，各フローが自身のアプリケーションと同じ性能要求を持つクラスタに正しく分類されているかを示す正確性に関する指標を定義し，比較を行う．

正確性を定義するにあたり，まず，各クラスタに以下の条件のいずれも満たすアプリケーションのラベルを付与する．

- 当該クラスタに属するアプリケーション種別のうち最多もの
- 当該クラスタに 3 つ以上の当該アプリケーション種別のフローが含まれている

上記の条件を満たすアプリケーションが存在しない場合，当該クラスタは適切なラベルを付けることができないクラスタであるとする．

その後，各フローについて当該フローのアプリケーション種別と同じラベルが付与されたクラスタに分類された割合を正確度として定義する．アプリケーション種別 i の正確度 C は以下のように定義される．

$$C(i) = \frac{f_{c_i}}{F_i} \tag{11}$$

F_i はアプリケーション種別 i のフローの総数，そのうち，正しくアプリケーション種別 i のクラスタに分類されたフロー数を f_{c_i} とする．

また，平均正確度 C を以下のように定義する．

$$C = \frac{f_c}{F} \tag{12}$$

ここで，F はフローの総数，f_c は正しく分類されたフロー数である．

4.3 比較対象

本評価では，AntTree をオンライン型のトラヒック識別への適用可能性を評価する．本評価では，以下のように AntTree を動作させる．まず，初期学習として，フロー 10 個のデータに対して，AntTree の手順を動作し，クラスタリングを行う．その後，フローの観測データを一つずつ追加しながら，クラスタリングを行う．フローのデータが一つ追加される度に，4 回ほどアリの探索を行うものとした．

また，本評価では，比較対象として，全フローの情報が得られた後に，K-means++法でクラスタリングを行った場合を用いる．

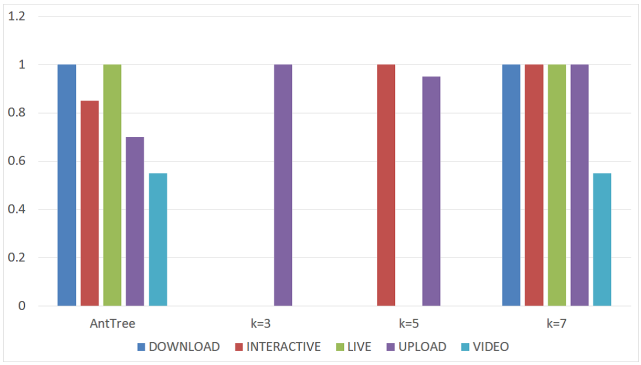

図 1　手法間の正確度の比較

4.4　評 価 結 果

図 1 に手法間の正確度の比較を示す．図より，いずれのクラスタリング手法を用いた場合でも Video on Demand に関する正確度が低いことが分かる．これは，Video on Demand フローの内ニコニコ動画のフローが統計情報において Download と同様の徴候を示しているためである．Video on Demand のアプリケーションは，帯域が空いていればより多くの動画データをダウンロードしバッファリングしようとするため，ネットワークに要求する性能要件は，Download と似ている．そのため，一部の Video on Demand のアプリケーションを Download と誤認してしまうことのネットワーク制御への影響は大きくないと考えられる．

クラスタリング手法間で正確度の比較を行うと，次々に到着するデータをオンラインで分類する手法である AntTree は，全フローの情報を得られたのちに K-means++でクラスタリングにおいて適切な k を定めた場合と同程度の正確度を達成できていることが分かる．AntTree では，各データが自律的に類似するデータを探索し，類似するデータが存在しない場合はサポートと繋がることにより新たなクラスタを構成することができる．その結果，順次データが到着する場合であっても，AntTree では，到着したデータに合わせて既存のクラスタに入れる，あるいは，新たなクラスタを構成するという判断を行うことができ，正確な分類ができている．つまり，AntTree は，順次到着するフロー情報の分類を正確に行うことができると考えられる．

5.　まとめと今後の課題

本稿では， Web アプリケーションによるトラヒックの分類に焦点を当て，トラヒック分類機に生物学の知見にもとづいたクラスタリング手法の評価を行った．評価の結果，生物学の知見に基づいた AntTree を用いることにより，次々と到着するフローをアプリケーションの種類ごとに分類することが可能であり，全フローの情報を取得後に K means++法で分類した場合と同程度の分類精度を達成することができることが明らかになった．

今後は正確度の向上や，ネットワーク制御と分類の連携について検討を行う予定である．

文　　献

[1] Takashi Miyamura, Yuichi Ohsita, Shin'ichi Arakawa, Yuki Koizumi, Akeo Masuda, Kohei Shiomoto, and Masayuki Murata, "Network virtualization server for adaptive network control," in *Proceedings of 20th ITC Specialist Seminar on Network Virtualization - Concept and Performance Aspects*, May 2009.

[2] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A survey on internet traffic identification," *Communications Surveys & Tutorials, IEEE*, vol. 11, pp. 37–52, Aug. 2009.

[3] L. Popa, A. Ghodsi, and I. Stoica, "Http as the narrow waist of the future internet," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pp. 6:1–6:6, 2010.

[4] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for qos: A statistical signature-based approach to ip traffic classification," in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, pp. 135–148, 2004.

[5] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, pp. 50–60, June 2005.

[6] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," in *Proceedings of the 2006 ACM CoNEXT Conference*, CoNEXT '06, pp. 6:1–6:12, 2006.

[7] A. Abraham, C. Grosan, and V. Ramos, **群知能とデータマイニング**. 東京電機大学出版局, July 2012. (栗原 聡, 福井 健一 : 訳).

[8] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, (Philadelphia, PA, USA), pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[9] H. Azzag, N. Monmarche, M. Slimane, and G. Venturini, "Anttree: a new model for clustering with artificial ants," *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, vol. 4, pp. 2642–2647, Dec. 2003.

[10] "Wireshark." http://www.wireshark.org.

[11] "Bitcasa." https://www.bitcasa.com.

[12] "Box." https://www.box.com.

[13] "Dropbox." https://www.dropbox.com.

[14] "Ustream." http://www.ustream.tv.

[15] "Youtube." http://www.youtube.com/.

[16] "ニコニコ動画." http://www.nicovideo.jp/.

[17] "Google マップ." https://maps.google.co.jp.

[18] "Yahoo!地図." http://map.yahoo.co.jp.