

修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報理工学研究科 情報・通信工学専攻 博士前期課程		
氏 名	中野隆介	学籍番号	1131089
論文題目	無線LANアクセスポイントの検索要求を用いた 屋内混雑度推定手法		
<p>要 旨</p> <p>ある場所がどの時間にどれだけ混雑しているか、という混雑情報はその場所の事業者にとっても利用者にとっても有益である。例えば、鉄道の混雑度（乗車率）を広く収集し分析を行えば空いている電車を優先的に検索する路線検索システムを作ることができるし、店舗で混雑度（来客者数）を収集すれば、来客者のうち何人が実際に商品を購入したかという分析を行うことができる。</p> <p>もっとも広く普及している混雑度の測定方法は目視測定であるが、人件費もかかり長期間データを収集することが難しいという欠点がある。混雑度測定の自動化方式としては、カメラを用いて導線解析する方式や二酸化炭素センサを用いる方式などがあるが、適用できるエリアが限定的であり、広く使われるには至っていない。</p> <p>そこで本研究では近年普及してきた無線LAN搭載モバイル端末に着目する。無線LAN端末は、周辺のアクセスポイントを検索するために不定期にプローブ要求をブロードキャストしている。この信号は端末が自然に送出するものであり、端末の特別な操作や事前の準備は必要ない。本研究では、この信号を利用し無線LAN信号のキャプチャ機器を設置するだけで混雑度の推定が行えるシステムを提案する。</p> <p>実際に鉄道列車内と大学の教室内で実験を行い、鉄道列車内では混雑度に対して相関係数0.74の特徴量を、教室内では在室人数に対して相関係数0.86の特徴量を得られることを示した。</p>			

平成 24 年度修士論文

無線 LAN アクセスポイントの
検索要求を用いた屋内混雑度推定手法

電気通信大学大学院情報理工学研究科
情報・通信工学専攻

学籍番号 : 1131089

氏名 : 中野隆介

指導教員 : 沼尾雅之 教授

副指導教員 : 岩崎英哉 教授

要旨

ある場所がどの時間にどれだけ混雑しているか、という混雑情報はその場所の事業者にとっても利用者にとっても有益である。例えば、鉄道の混雑度（乗車率）を広く収集し分析を行えば空いている電車を優先的に検索する路線検索システムを作ることができるし、店舗で混雑度（来客者数）を収集すれば、来客者のうち何人が実際に商品を購入したかという分析を行うことができる。

もっとも広く普及している混雑度の測定方法は目視測定であるが、人件費もかかり長期間データを収集することが難しいという欠点がある。混雑度測定の自動化方式としては、カメラを用いて導線解析する方式や二酸化炭素センサを用いる方式などがあるが、適用できるエリアが限定的であり、広く使われるには至っていない。

そこで本研究では近年普及してきた無線 LAN 搭載モバイル端末に着目する。無線 LAN 端末は、周辺のアクセスポイントを検索するために不定期にプローブ要求をブロードキャストしている。この信号は端末が自然に送出するものであり、端末の特別な操作や事前の準備は必要ない。本研究では、この信号を利用し無線 LAN 信号のキャプチャ機器を設置するだけで混雑度の推定が行えるシステムを提案する。

実際に鉄道列車内と大学の教室内で実験を行い、鉄道列車内では混雑度に対して相関係数 0.74 の特徴量を、教室内では在室人数に対して相関係数 0.86 の特徴量を得られることを示した。

目次

第 1 章	はじめに	1
1.1	背景	1
1.2	本論文の構成	3
第 2 章	IEEE802.11 無線 LAN	4
2.1	802.11 の用語	4
2.2	ネットワークの種類	5
2.3	ステーションが AP に接続するまでの流れ	6
2.4	スキャンの種類	6
第 3 章	関連研究	9
3.1	電波の到着時間差を利用した手法 (TDOA 方式)	9
3.2	RSSI 値を利用した手法	9
3.3	既存研究との違い	13
第 4 章	提案手法	14
4.1	混雑度の定義	14
4.2	測定環境のセットアップ	16
4.3	混雑度推定手法の流れ	16
4.4	位置推定手法	17
4.5	端末サンプリング	20
4.6	時間補完アルゴリズム	22
4.7	混雑度推定	23
第 5 章	評価実験	24
5.1	電車型の混雑度推定	24
5.2	公園型の混雑度推定	31
第 6 章	考察	43
6.1	端末サンプリング	43
6.2	電車型の混雑度推定	44
6.3	公園型の混雑度推定	45

第 7 章	おわりに	46
7.1	まとめ	46
7.2	今後の課題	46

第1章 はじめに

1.1 背景

ある場所がどの時間にどれだけ混雑しているか、という混雑情報はその場所の事業者にとっても利用者にとっても有益である。例えば、鉄道の混雑度（乗車率）を広く収集し分析を行えば空いている電車を優先的に検索する路線検索システムを作ることができるし、店舗で混雑度（来客者数）を収集すれば、来客者のうち何人が実際に商品を購入したかという分析を行うことができる。

しかし、混雑度の測定は非常に面倒である。もっとも広く普及している混雑度の測定方法は目視測定であるが、人件費もかかり長期間データを収集することが難しい。混雑度測定の自動化方式としては、カメラを用いて導線解析する方式 [23] や二酸化炭素センサを用いる方式 [25] などがあるが、適用できるエリアが限定的であり、広く使われるには至っていない。鉄道会社も独自に列車の混雑度情報を収集しているが、首都圏の鉄道会社の7割が目視による測定を利用している [19]。



図1 NAVITIME 社の電車混雑レポート [6]

目視による測定をクラウドソースで大量に行おうとしているサービスも存在する。NAVITIME の電車混雑レポートサービス（図1）は、ユーザが目視で混雑度を入力してデータを収集している。しかしこれは非常に面倒な作業である。電車というのは駅

に停車する度に混雑度が変化するものだから、正しいデータを収集するには各駅毎にユーザが混雑度を再入力しなければならない。だが実際に投稿されている様子を観察すると、1回の乗車で複数の混雑情報が投稿されることはまずないようである。

NAVITIME のサービスからわかるように、混雑度情報には確かな需要があるものの、一般に収集が難しい。そこで本研究では近年普及してきた無線 LAN 搭載モバイル端末に着目する。無線 LAN 端末は、周辺のアクセスポイントを検索するために不定期にプローブ要求をブロードキャストしている。この信号は端末が自然に送出するものであり、端末の特別な操作や事前の準備は必要ない。この信号から混雑度を推定することができれば、無線 LAN 信号のキャプチャ機器を設置するだけで混雑度の推定が行えるシステムを実現できる。

本研究ではモバイル端末が送出するプローブ要求から混雑度推定を行うシステムを構築することを目的とする。

1.2 本論文の構成

第2章

無線 LAN の標準規格、IEEE802.11 について述べる。

第3章

無線 LAN を用いた位置推定に関する関連研究を紹介する。

第4章

提案する混雑度推定システムについて述べる。

第5章

提案システムに対する評価実験を行った結果を報告する。

第6章

実験に対する考察を述べる。

第7章

まとめと今後の課題を述べる。

第 2 章 IEEE802.11 無線 LAN

無線 LAN 規格は IEEE802.11 で定められている。本章では 802.11 のうち、本論文に関連する箇所を抜粋して紹介する。本章の内容は文献 [9] に基づいている。

2.1 802.11 の用語

802.11 のネットワークは 4 つの主要な物理的コンポーネントからなる。

ステーション

無線 LAN ネットワークインターフェースを持つコンピュータ。802.11 はステーション間でデータを転送するための規格である。ステーションの代表格としては、ノート PC とスマートフォンがある。

アクセスポイント (AP)

802.11 ネットワーク上のフレームを残りの世界に配信するためのゲートウェイ。一般的には無線ネットワークと有線ネットワークをブリッジする装置。アクセスポイントの代表格としては、無線 LAN ルータがあるが、スマートフォンのテザリング機能やモバイルルータの普及によって、非常に多様なアクセスポイントが用いられている。

無線媒体

フレームを実際に転送する時に用いる物理層。802.11 では複数の物理層を開発することを認めており、当初は 2 つの無線 (RF) の物理層と 1 つの赤外線物理層が標準化されていた。現在では RF による物理層のほうがはるかに普及している。本論文では物理層に RF を用いた無線 LAN システムのみを扱う。

ディストリビューションシステム

複数のアクセスポイントによって広域の単一なネットワークが構築されている場合、アクセスポイント間でフレームの伝送を行わなければならない。ディストリビューションシステムはそのフレーム伝送の方法のことで、実用には必須だが 802.11 では詳細は規定されていない。無線 LAN ルータではバックボーンにイーサネットがよく用いられる。また、バックボーンに無線を用いる WDS (Wireless Distribution System) も存在する。

2.2 ネットワークの種類

802.11 では 2 種類のネットワークが規定されている。図 2 に概観を示す。

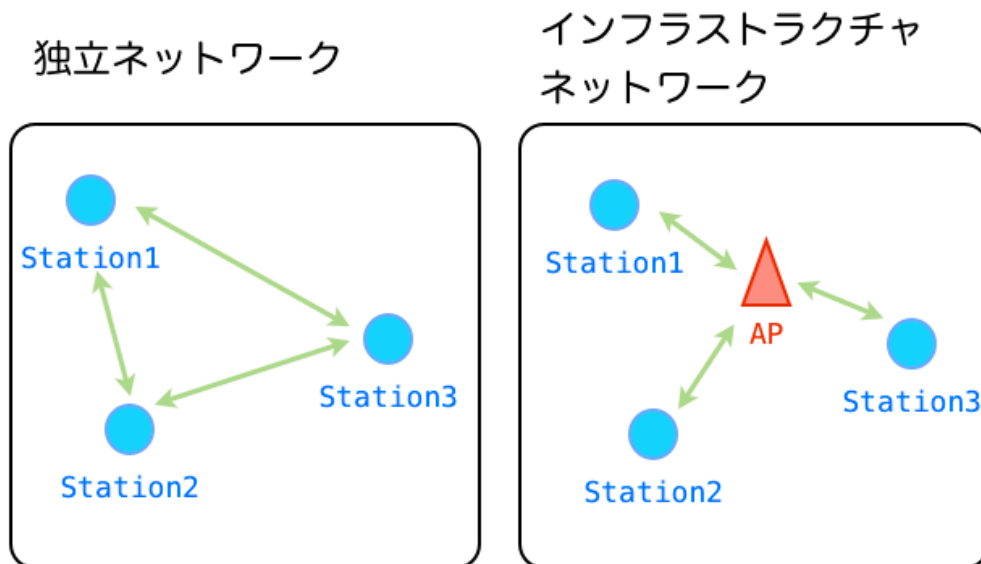


図 2 802.11 で規定されている 2 つのネットワーク

2.2.1 独立ネットワーク

独立ネットワークでは、ステーションが互いに直接通信する。したがって、ステーションは互いに直接通信可能な範囲にいないといけない。現在は PC 間の通信に用いられることは少なくなったが、情報家電間の通信や携帯ゲーム機の通信などに用いられている。短時間のみ存在する独立ネットワークはアドホックネットワークとも呼ばれる。

2.2.2 インフラストラクチャネットワーク

インフラストラクチャネットワークはアクセスポイントを使用する点が独立ネットワークと異なる。アクセスポイントは、同一ネットワーク内の全ての通信に使われる。ステーション間の通信は必ずアクセスポイントを介して通信するので、2 ホップの通信が必要になる。インフラストラクチャネットワークは現在広く普及しており、一般的に無線 LAN ネットワークと言えばこちらのネットワーク形態を指す。このネット

ワーク形態はノート PC やスマートフォン等のインターネットに接続する必要がある機器で用いられる。本論文ではインフラストラクチャネットワークのみを扱う。

2.3 ステーションが AP に接続するまでの流れ

インフラストラクチャネットワークにおいて、ステーションがアクセスポイントに接続し通信可能な状態になるには、アクセスポイントのスキャン、認証、アソシエーションの 3 ステップを踏む必要がある。

ステーションはまず接続可能なアクセスポイントと利用可能なチャンネルを検索するため、スキャンを行う。スキャンによって周辺のアクセスポイントの SSID（ネットワークの名前）、BSSID（MAC アドレス）、使用しているチャンネル、暗号化方式、電波受信強度などの情報を入手する。次に接続するアクセスポイントの選択を行う。これは人間がアクセスポイントのリストから選択する実装もあれば、もっとも電波受信強度が強いものに接続する実装もありうる。接続するアクセスポイントを決定したら、ステーションとアクセスポイントの間で認証を行う。ここで言う認証とは MAC アドレスに基づいたもので、一般的にはこの認証の後、WEP や WPA 等の暗号化を伴うユーザ認証を行う。認証が完了すると、アソシエーションが行われる。これは、このステーションはこのアクセスポイントに接続している、という情報を登録するもので、これによってはじめてステーションへのフレーム配信が可能になる。

本論文ではステーションが行うスキャンに着目する。

2.4 スキャンの種類

ステーションが行うスキャンにはパッシブスキャンとアクティブスキャンの 2 種類がある。

2.4.1 パッシブスキャン

パッシブスキャンは信号の送出を行わないでスキャンを行う。パッシブスキャンでは、ステーションは使用可能な全てのチャンネルそれぞれでビーコンフレームの到着を待つ。ビーコンを受信すると、そこからそれらを送出したアクセスポイントに関する情報を取り出す。ステーションが信号を送出せずともスキャンが完了できるように、ビーコンフレームにはネットワークに接続するために必要な情報が全て含まれている。

ビーコンフレームのフォーマットを図 3 に示す。

2.4.2 アクティブスキャン

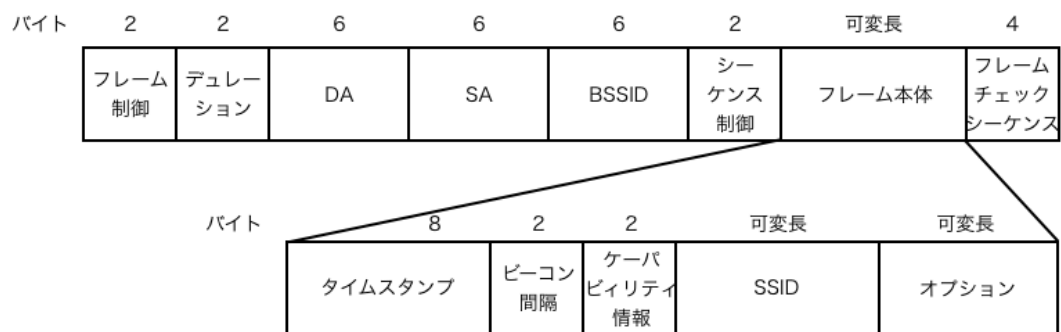


図3 ビーコンフレームのフォーマット。文献 [9] より引用。

アクティブスキャンでは、使用可能な各チャンネルで、アクセスポイントからの応答を得るためプローブ要求フレームが用いられる。アクティブスキャンはアクセスポイントからのアナウンスが到着するのを待つのではなく、ステーションが自分自身でアクセスポイントを発見しようとする。アクティブスキャンを行うステーションは使用可能な全てのチャンネルについて、以下に示す処理を行う。

1. 指定されたチャンネルに移り、新たなフレームが到着するか、あるいは Probe-Delay タイマが時間切れになるのを待つ。到着フレームが発見されれば、そのチャンネルは使用されているのでプローブを行えることがわかる。発見されない場合はこのチャンネルのスキャンをスキップする。
2. プローブ要求フレームをブロードキャストで送信する。
3. MinChannelTime が経過するのを待つ。
 - a. 無線媒体がビジーにならないければ、そこにはネットワークがない。次のチャンネルに移る。
 - b. MinChannelTime の間に無線媒体がビジーになれば、MaxChannelTime が経過するのを待ち、その間に受信したプローブ応答フレームの処理を行い、次のチャンネルに移る。

ここで ProbeDelay, MaxChannelTime, MinChannelTime はパラメータで、多くの場合デバイスドライバが決定する。

アクティブスキャンで送出されるプローブ要求フレームのフォーマットを図4に示す。

フレームに含まれる DA フィールドは宛先 MAC アドレスで、プローブ要求はブロードキャストなため FF:FF:FF:FF:FF:FF になっている。SA フィールドは送信元 MAC アドレスで、ステーションのアドレスが書かれている。シーケンス制御フィー

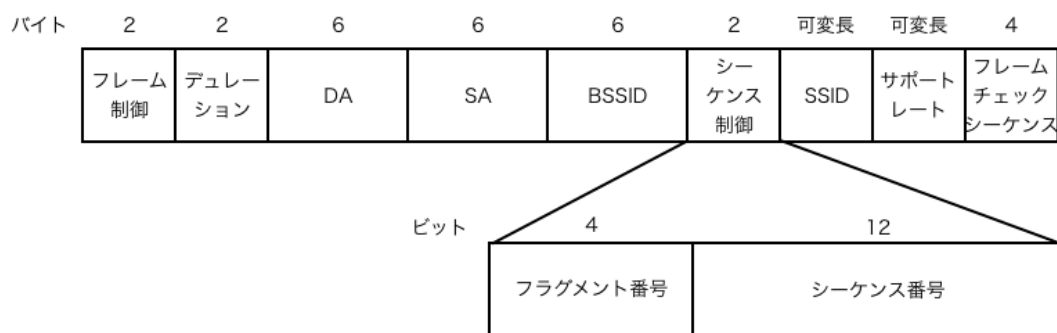


図4 プローブ要求フレームのフォーマット。文献 [9] より引用。

ルドにはフラグメント番号サブフィールドとシーケンス番号サブフィールドが含まれている。シーケンス番号は、伝送されるフレーム番号の 4096 の剰余で計算される。シーケンス番号は 0 から始まり、フレームが伝送される度に 1 増加する。伝送するパケットがフラグメント化する場合には全てのフラグメントに同一のシーケンス番号がつく。異なるフラグメントではフラグメント番号が異なる。

このフレームフォーマットで重要なことは、DA フィールドがブロードキャストなプローブ要求は誰でも受信して良く、更に SA フィールドにはステーションのアドレスが含まれているため、プローブ要求をキャプチャすることで、周辺でどんなステーションがアクティブスキャンを行ったかがわかる、ということである。

第3章 関連研究

現在広く普及している無線通信方式 IEEE802.11 は、以前から GPS に代わる位置測位手段として研究されてきた。その位置推定手法は大きく分けて2つの手法、電波の到着時間差を利用した手法 (TDOA 方式) と RSSI 値を用いる手法が提案されている。

3.1 電波の到着時間差を利用した手法 (TDOA 方式)

TDOA (Time Difference of Arrival) 方式は GPS の測位原理と似ていて、端末が送信した信号を各アクセスポイントで受信する際の時刻の差から端末の位置を推定する手法である。屋外では非常に高い精度が期待できるが、信号は光速で伝搬するためアクセスポイント間の時刻をナノ秒オーダーで同期しておく必要がある。しかし屋内では反射波や錯乱波の影響を受けやすく、安定した精度を得ることが難しい。また、正確に時刻同期を行うために特別な装置が必要であり、導入コストが高い。

TDOA 方式で実用化されたものとして日立的 AirLocation[20] がある。AirLocation ではアクセスポイントをマスター 1 台、スレーブ 3 台で構成し、マスターを時刻同期用に使用することで同期の問題を低減している。位置測位を行う際はまずマスターが測定端末とスレーブへ同期信号を送信する。次に測定端末がスレーブへ信号を送信し、各スレーブはマスターとの同期信号との時刻差を用いて位置測位を行う。これによって 1~2m 程度の精度が実現できることが示されている。しかしアクセスポイント側、端末側の両方に事前の準備が必要であり、広く使用されるには至っていない。

3.2 RSSI 値を利用した手法

RSSI (Receive Signal Strength Indicator: 受信信号強度) 値は端末とアクセスポイントが離れば離れるほど小さくなる。一般に RSSI 値と距離との関係は非線形で、RSSI は距離の二乗に反比例することが知られている。この減衰関係をモデル化し、位置推定を行うのが RSSI を用いる手法である。

ここではアクセスポイントが送信するビーコンを用いてステーション側で位置推定を行う手法 (図 5) と、ステーションが送信するプローブ要求を用いてアクセスポイント側で位置推定を行う手法 (図 6) の 2 つに分類して紹介する。

3.2.1 ビーコンを用いて位置推定を行う研究

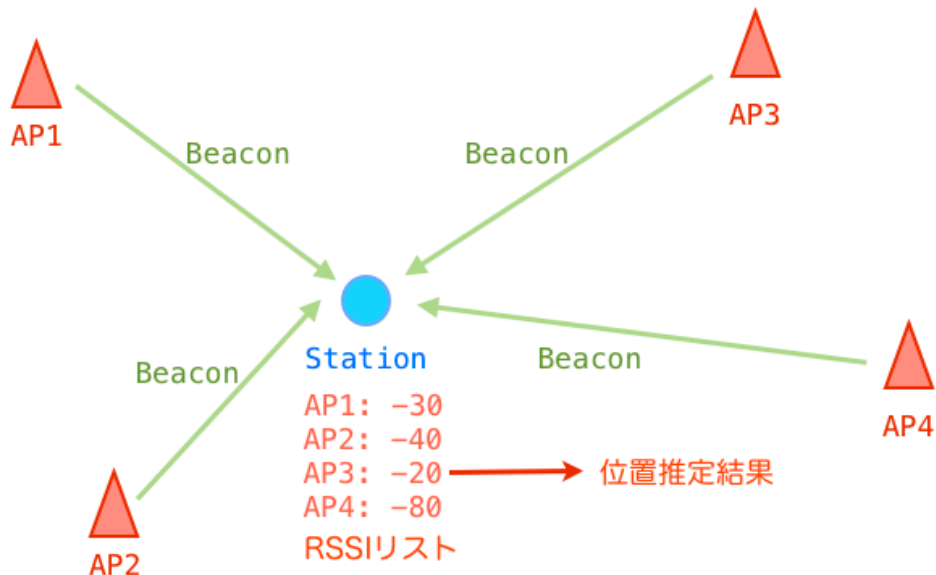


図5 ビーコンを用いた位置推定。各アクセスポイントからの信号をステーション内で集約する。

Microsoft の RADAR[8] は無線 LAN 測位の先駆けとなった研究で、建物内の直線廊下上で無線 LAN 信号をキャプチャし、その RSSI 値から三角測量を行うことで 3m の分解能で位置測位が可能であると示している。RADAR の手法は三角測量の原理に基づいていることから、Triangulation とも呼ばれる。

Triangulation は屋外や直線上などの反射波や散乱波の影響を受けにくい場所では高い精度を期待できるが、壁や部屋を挟んだ場所の位置推定には向かない。Triangulation を採用した他の研究としては LaMarca ら [13] のものがある。LaMarca らは GPS ログを記録しながら街中で信号キャプチャを行い、そこから得た大量の RSSI 値情報から、アクセスポイントの位置制約条件を抽出しアクセスポイントの位置を推定、その結果からアクセスポイント DB を更新する手法を提案している。

Krumm らの NearMe Server[11] はビーコンを用いておおまかな位置推定を行うアプリケーションで、無線 LAN 信号間の類似度を定義し、その距離を利用することで位置推定を行っている。アプローチとしては機械学習における k-Nearest Neighbor 法に非常によく似ている。NearMe Server の手法は Proximity とも呼ばれる。

Proximity は位置推定精度は低いが、位置推定システムの構築が容易で、アクセス

ポイントが追加された場合でも推定システムのアップデートも簡単であるという特徴がある。

伊藤ら [17] は位置推定を行う各場所で前もって RSSI 値を観測しておき、そこからベイズ推定のアプローチによって位置推定を行う手法を提案している。このような機械学習的アプローチを用いた手法を Scene Analysis と呼ぶ。

Scene Analysis は、位置推定を行う領域中の複数の地点でシーン（RSSI 値と補助的な情報）を観測しておき、事前に分かっている各シーンの測定位置から機械学習を行い位置推定を行う方式である。Scene Analysis と前述の 2 つの手法とのもっとも大きな違いは、位置推定にアクセスポイントの位置情報を全く必要としない点である。あるシーンであるアクセスポイントからの信号をある強度で受信した、という情報を記録し、その信号を送信したアクセスポイントがどこにあるかという情報は使用しない。そのため、Scene Analysis では他人が設置しているアクセスポイントを勝手に利用して位置推定モデルを構築することができる。事前にシーンの学習を行わなければならないが、他人のアクセスポイントを使用できるため非常に多くの情報を利用でき、RSSI 値を用いた位置推定手法としてはもっとも高い精度が期待できる。

このシーン情報の収集をクラウドソースで行おうとしているプロジェクトも存在する。PlaceEngine[26] や PlaceLab[12]、Locky[18] などはユーザーが所持している端末でシーンの観測を行わせ、位置情報と共にサーバへ送信させることで、広範囲の無線 LAN 測位システムの構築を目指している。Google や Apple はこのアプローチを更に大規模化しており、彼らの提供するスマートフォン（Android と iOS）には OS レベルでシーン収集機能が組み込まれている。彼らはスマートフォンに搭載されている Wi-Fi と GPS を併用してシーン情報収集を自動化し、その成果として位置推定 API を公開している [1][2]。

Scene Analysis は反射波や散乱波の影響を含んだ学習器を作成できるため屋内でも有効な手法であり、藤田ら [22] は無線 LAN の電波伝搬特性を正規分布に近似し、建物内の電波強度分布を Gaussian Mixute Model によってモデル化することで、事前の学習に必要なデータを大幅に減少させる手法を提案している。

3.2.2 プローブ要求を用いて位置推定を行う研究

Interlink Networks のテクニカルレポート [14] でプローブ要求を用いてアクセスポイントに接続していないステーションを発見し、追跡可能であることが示されている。テクニカルレポートが発表された当時は無線 LAN 機器はあまり普及しておらず、無線 LAN と言えばノート PC と無線 LAN カードの組み合わせが常識であった。しかしこの 10 年で状況は大きく変化し、スマートフォンや携帯ゲーム機など、無線 LAN を搭載するモバイル端末が広く普及した。これらの端末が送出するプローブ要求から

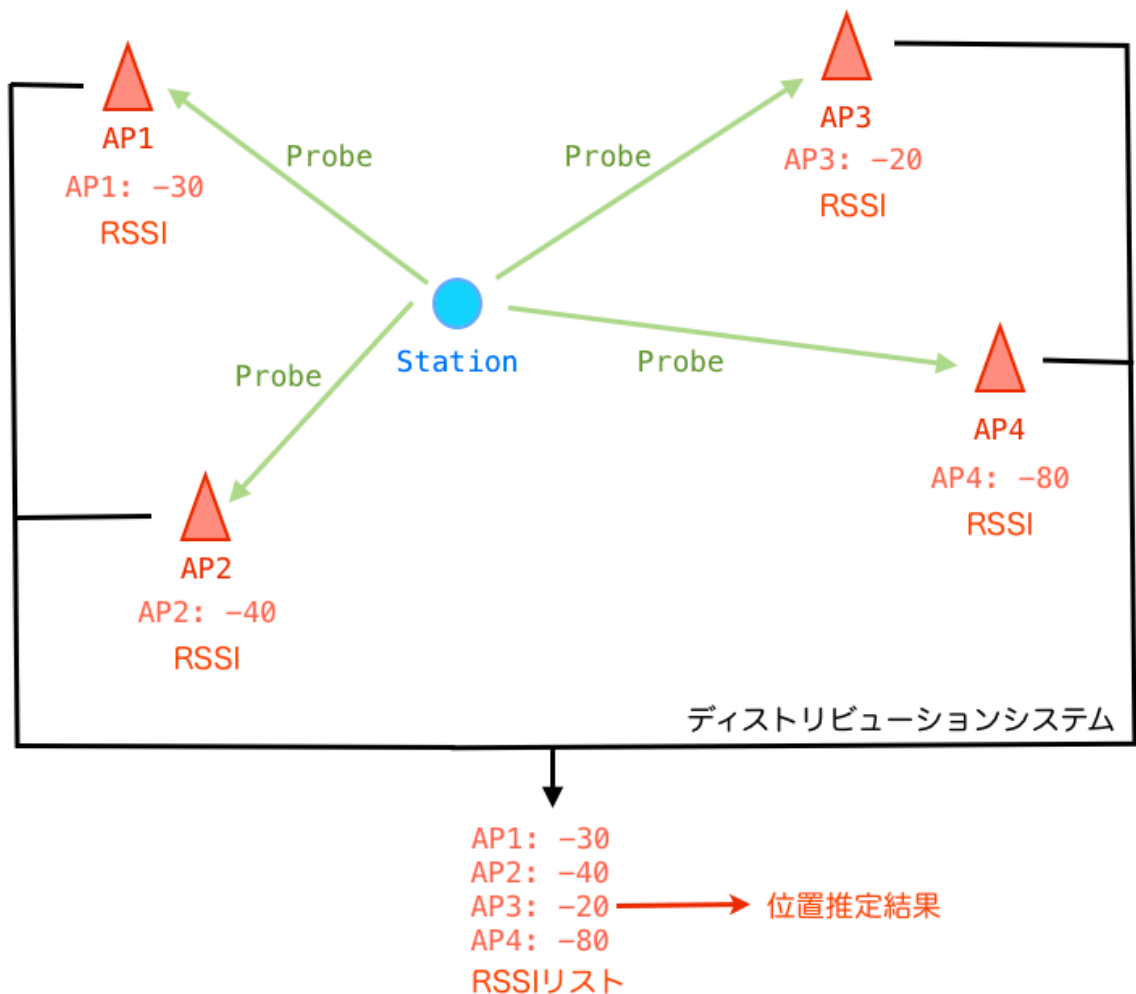


図 6 プローブ要求を用いた位置推定。各アクセスポイントでの測定結果をディストリビューションシステム越しに集約する必要がある。

位置推定が行えれば、その持ち主を追跡することができるため、プローブ要求による位置推定が再び研究されている。

阿瀬川ら [15][16] はステーションが送出するプローブ要求を複数のアクセスポイントで受信し位置推定する環境を提案し、そこで予想される位置推定誤差の問題について述べている。特に大きな問題は、プローブ要求をステーションが利用可能な全てのチャンネルで送出されるため、プローブ要求をキャプチャしているチャンネルの隣のチャンネルへ送出されたものも受信してしまうことである。隣のチャンネルへ送信されたプローブ要求はバンドパスフィルタのパラメータが異なるため RSSI が低く観測される。この RSSI も含めて位置推定を行うと、推定誤差が大きくなる。そのため阿瀬川らはプローブ要求を RSSI の値で k-means クラスタリングし、RSSI が最も高い

クラスタの平均値を用いることで他チャンネルによる影響を排除している。

また、プローブ要求特有ではないが、多数の端末の位置推定を行う際の問題として端末のアンテナの性能差がある。これに対しては Kjærgaard ら [10] が複数のアクセスポイント間での RSSI 比を用いる手法を提案しているほか、片山ら [24] は Kjærgaard らの手法を拡張し、測定時と学習時の RSSI 値の違いを正準相関分布を用いて補正し、測定端末の違いによる影響を少なくする手法を提案している。

端末のアンテナの違いを吸収するために RSSI の差を用いる手法は、ビーコンを用いた位置推定では広く使われている。しかし、プローブ要求を用いると実現するのが難しい。これは時刻同期の問題に起因する。ビーコンを用いた手法では、周辺のアクセスポイントが送信するビーコンを端末内で同期、すなわち受信したタイミングで比を計算すれば良い。しかしプローブ要求を用いると、端末が送信したプローブを周辺のアクセスポイントで受信、同期を行うのでアクセスポイント間の時刻同期が必要になる。これは 0.1 ミリ秒オーダーの時刻同期が必要となり、ディストリビューションシステムにイーサネットを採用したシステムでは実現が難しい。現在、この問題に対する解決手法は示されていない。

3.3 既存研究との違い

本論文ではプローブ要求を用いて位置推定を行い、その結果から混雑度を推定する手法を提案する。また、それに付随してプローブ要求を用いて位置推定を行う際に問題となっている、RSSI 比計算のための時刻同期手法を提案する。既存研究とは位置推定後の集約を扱っている点が異なる。

第4章 提案手法

802.11 におけるプローブ要求はブロードキャストされ、そのフレームには送信元 MAC アドレスを含んでいる。現在 802.11 方式の通信機器を搭載した端末は広く普及している。例えばほとんどのスマートフォンは Wi-Fi を搭載している。スマートフォンは携帯電話を置き換えようとしているものであるから、多くの人が携帯して持ち歩くという特徴がある。

つまり、スマートフォン等のモバイル端末が送出するプローブ要求は人そのものが送出していると捉えることができ、それを用いて人流や混雑度が計測可能であると考えられる。本章ではプローブ要求を用いてある場所の混雑度を推定する手法を提案する。

4.1 混雑度の定義

本論文では、混雑度を調査したい場所のことを**観測場所**と呼ぶ。観測場所とはある空間の一部であり、観測場所とその他の空間とを分ける境界 B が存在する。いま空間上を移動する人 P が居たとき、 P が観測場所の外から境界 B を越えて観測場所へ移動してくることを P が入室したと言い、 P が B を超えて外へ移動していくことを P が退室したと言うことにする。時刻 t において観測場所内にいる (t より前の時刻に入室してまだ退出していない) 人の数のことを在室人数 $N(t)$ と呼ぶ。

このとき、観測場所の混雑度 $C(t)$ を、時刻 t によって変化する在室人数 $N(t)$ と比例した量として定義する。

$$C(t) \propto N(t) \quad (1)$$

混雑度 $C(t)$ は必ずしも在室人数である必要はなく、「空いている」「混んでいる」等の順序付きの質的な量でも良い。時刻 t は連続値ではなく離散値とする。この時刻 t の分解能の大きさは混雑度推定手法を適用するユーザが決定する。この分解能を rt と呼ぶ。分解能 rt が大きいほど混雑度推定は容易になる。本論文では混雑度推定を機械学習問題に帰着し、推定に用いる特徴量の計算手法を提案する。

また、一度の観測の単位を**系列**と呼ぶことにする。一つの系列は一度観測を開始してから終了するまでに観測したデータ列を示す。混雑度の計算は一つの系列の観測を終了する度にバッチ処理で行い、リアルタイム計算は本論文では扱わない。本論文で扱う混雑度は、系列毎に独立で計算し、以前の系列には影響されないと仮定する。加

えて、系列が切り替わるタイミングで観測場所の在室人数が一度ゼロになると仮定する。例えば、店舗の場合は一日の営業時間を一つの系列として、列車の場合は始発駅から終点駅までを一つの系列とすればこの仮定を満たせる。

本論文では混雑度の推定精度をより向上させるため、観測場所を3つの種類（表1）に分けて議論する。

表1 観測場所の種類

	入退室制約あり	入退室制約なし
場所移動あり	電車型	-
場所移動なし	映画館型	公園型

場所移動のありなしは観測場所が地理的に移動するかどうかを示している。ここでの移動とは境界の平行移動のみを許す。例えば鉄道の列車の車両内を観測場所とするなら、境界は各車両の外壁であり、列車の運行に合わせて平行移動している。このケースは場所移動ありのクラスになる。鉄道のホームを観測場所とするなら、その境界は移動しないので、場所移動なしのクラスになる。公共交通機関を観測場所とするなら多くは場所移動ありのクラスになる。

入退室制約のありなしは、人の出入りが時間によって制限されているかどうかを示している。具体的には鉄道の車両内は駅の停車中でなければ乗り降りできないため、入退室制約がある。公園や店舗などの場所では人の出入りが自由に行えるため、入退室制約はない。直感的には、観測場所のドアが全て閉まる状態が存在するのが入退室制約ありで、ドアが常に開いている観測場所は入退室制約なしになる。つまり、ある時間内で在室人数の変化がないと保証された場所が入退室制約ありのクラスになる。

入退室制約は以下の式で表現できる。観測場所に、その場所内外を移動できるただ1つの仮想的なドアがあったとする。時刻 t の観測場所の在室人数を $N(t)$ とする。時刻 tc_i から時刻 to_i までドアが閉まっていて、その間に一度も開くことはないとする、その間 $N(t)$ は一定なので、入退室制約 T_{door} は間隔 (tc_i, to_i) の集合だと考えることができ、

$$T_{door} = \{(tc_i, to_i) \mid tc_i \leq t < to_i, N(t) = \text{const.}\} \quad (2)$$

と表現できる。以降の議論の簡単化のために、 $tc_i < tc_{i+1}$ を満たすとする。この式から、ドアが開いている時間は to_i から tc_{i+1} まで、閉まっている時間は tc_j から to_j と表現できる。

入退室制約がある場合は、ドアが閉まっている時間のみを混雑度推定の対象とする。一般に入退室制約が存在する環境では、ドアが開いている時間よりも閉まっている時

間のほうが長く、ドアが閉まっている時間帯の混雑度情報のほうが有益である。また、そのような環境ではドアが開いている時間帯の人の出入りが非常に激しく、混雑度の正解データを作ることが難しい。そのため入退室制約がある場合はドアが閉まっている時間 tc_j から to_j のみの混雑度を対象とする。

4.2 測定環境のセットアップ

本手法ではプローブ要求のキャプチャに用いる機器を観測機器と呼ぶ。場所移動を伴うかどうかで必要な観測機器の数が異なる。一般的に使用される無線 LAN 機器の信号到達距離の目安は見通し 100m であるので、これを基準に測定機器の必要数を決定する。

4.2.1 場所移動ありの場合

観測場所内をカバーする位置に観測機器が 1 つあれば良い。ほとんどのケースにおいて、観測機器は 1 つで、測定場所の中央に設置すれば十分だと考えられる。1 つで観測場所内をカバーしきれない場合は複数必要だが、1 つで半径 100m のエリアをカバーできるため、フェリーのような巨大な建造物を除けば、1 つで十分だと考えられる。

4.2.2 場所移動なしの場合

観測場所内の内外の位置推定を行うために観測機器が複数必要である。観測場所が 100m 四方以下のエリアの場合、観測場所を取り囲むように 3 つ以上の観測機器が設置できれば十分である。それ以上のエリアの場合、観測場所の各点でプローブ要求を送信し、常に 3 箇所以上の観測機器で受信できるように設置する。

4.3 混雑度推定手法の流れ

混雑度推定手法の流れを図 7 に示す。

まず設置した観測機器でプローブ要求をキャプチャし、一つの系列分保存しておく。次に収集した端末（ステーション）の位置推定を行い、観測場所の外部にあると推定されるデータを削除する。位置推定後は受信した端末から一部をサンプリングし、その端末を時刻別に合計することで特徴量とする。最後にその特徴量を用いて機械学習を適用し、混雑度を推定する。

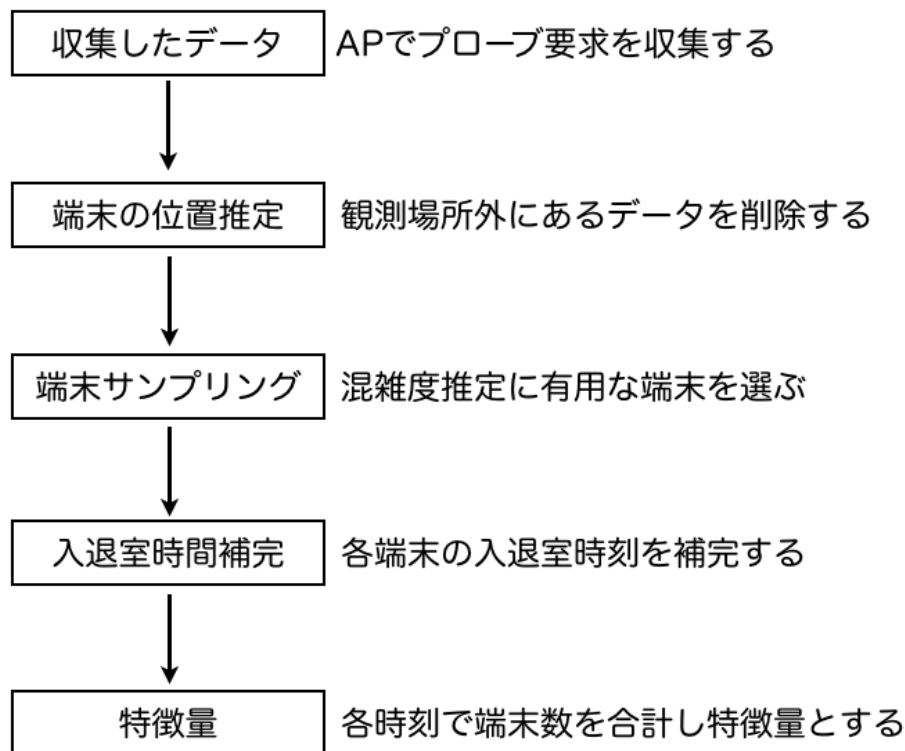


図 7 混雑度推定の流れ

4.4 位置推定手法

プローブ要求は観測機器の設置位置の約半径 100m から受信する。多くの場合、この受信エリアは観測場所より大きいいため、プローブ要求を送出した端末が観測場所の中にいるのか外にいるのかを位置推定する必要がある。

4.4.1 場所移動ありの場合

観測場所が移動しているのだから、観測場所内にいる端末も同じく移動していて、観測場所外にいる端末は移動していない。つまり、移動中にもかかわらず観測し続ける端末は場所内にいたと位置推定を行える。この推定のためにはいつ移動中かを知る必要があるが、これには入退室制約が利用できる。場所移動ありの場合の入退室制約とは移動中・停止中の時刻を示す時刻表とほぼ等価である。

また、位置推定の簡単化のために、ひとつの系列中で端末は一度しか観測場所の内外を移動しないと仮定する。例えば鉄道の場合、ある列車を駅 A で降車した端末が再

び駅 B で乗車することはない、とする。この仮定により、ひとつの系列中の端末の位置は時系列で計算する必要がなくなる。

以上から、場所移動ありの場合の位置推定アルゴリズムが求まる。ある系列でプローブ要求を観測している時に、全ての端末 D_k 毎にその端末をはじめて受信した時刻 ts_k と最後に受信した時刻 $te_k (\geq ts_k)$ を記録しておく。 $to_i < ts_k$ かつ $te_k \leq tc_{i+1}$ のとき、端末 D_k は入退室可能の状態のみで受信しており、移動中には受信していない。ここから D_k は観測場所の外側にいたと位置推定が行える。

また、 $tc_i \leq ts_k$ かつ $ts_k < to_i$ のとき、 D_k は入退室不可（移動中）に受信した端末となる。このとき、 $td_k = te_k - ts_k$ を定義し、 $td_k < dt$ なら観測場所の外側にいたと位置推定する。 dt は移動の平均速度から計算する。観測場所が平均秒速 v [m / s] で移動している時、観測機器の 100m 前後まで受信できるのだから、移動中に観測場所の外から最大 $200/v$ 秒の間受信することがある。よって $dt = 200\alpha/v$ とする。 α は安全係数で、2 から 5 程度を設定しておく。

この手法によって場所外にいと推定されなかった端末全てを場所内にいたとする。このアルゴリズムの概念図を図 8 に示す。水平の線分はそれぞれ各端末の ts_k と te_k を結んだもの。

この手法の利点は、位置推定を行うのに電波受信強度（RSSI）を必要としない点である。送信元端末によって送出電波強度は異なるため、RSSI を利用する際はその個体差を吸収するモデルを構築しなければいけないが、この手法ならば送出出力が異なる端末であってもそのまま位置推定を行える。

4.4.2 場所移動なしの場合

無線 LAN を用いた通常的位置推定手法である、電波強度を用いた手法を用いる。本手法では Scene Analysis のアプローチを採用する。最終的に位置推定の結果は観測場所の外か中かだけなので、クラス分類問題に近い Scene Analysis が適用しやすい。

プローブ要求を用いた位置推定に関する問題として、RSSI の不安定性が知られている。プローブ要求は端末が利用可能な全てのチャンネルに送出を行うため、測定機器でキャプチャしているチャンネルの隣のチャンネルへ送出されたプローブも受信してしまう。隣のチャンネルで受信したプローブ要求は RSSI が低くなるため、その RSSI 値を含めて位置推定を行うと推定誤差が大きくなる。

また、多様なモバイル端末を位置推定の対象とするため、各端末の電波送出出力が異なることも考慮しなければならない。この問題に対する広く知られているアプローチは各観測機器で受信した RSSI 値の比を用いることである。しかし前述の通り、プローブ要求では RSSI 値の異なるパケットが複数混ざることになるので、比を用いる手法もそのまま適用すると推定誤差が大きくなると考えられる。

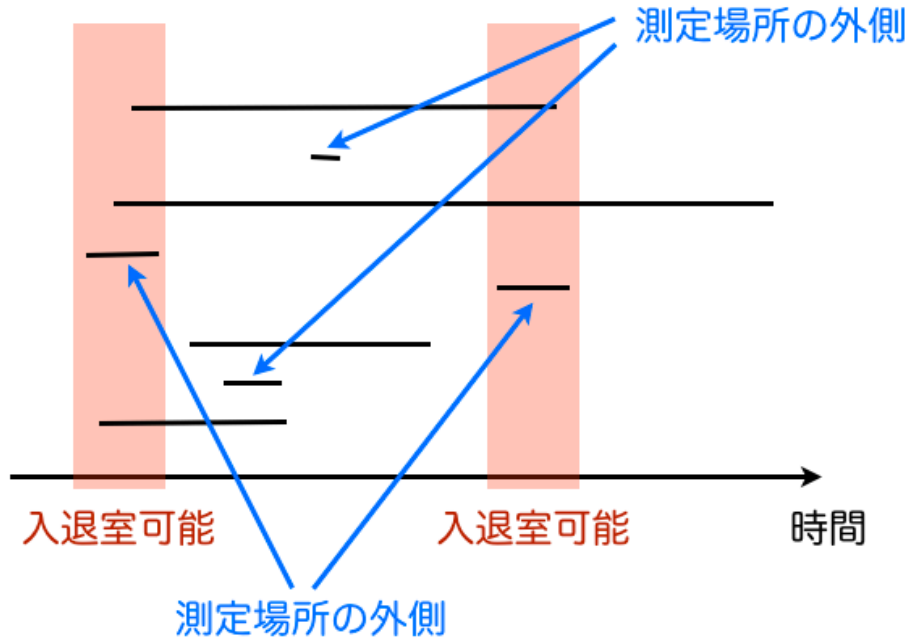


図 8 場所移動ありの場合の位置推定アルゴリズム

この問題に対する解決策は、各観測機器間で時刻同期を行い、各地点で受信したパケットを同期することだが、各観測機器間で 0.1 ミリ秒オーダーの時刻同期をし続けなければならない、ディストリビューションシステムにイーサネットや WDS を用いたシステムでは構築が難しい。

そこで、プローブ要求のシーケンス番号を用いるアプローチを提案する。プローブ要求フレームにはシーケンス制御サブフィールドが存在し、各チャンネルをホップする際にシーケンス番号が増加する。各観測機器で受信したプローブ要求をこのシーケンス番号を用いて同期することで、より正確な RSSI の信号比が求まると考えた (図 9)。

シーケンス番号同期のアルゴリズムを Algorithm 1 に示す。一定時間の間にある端末から受信したプローブ要求の数を n とする。 i 番目に受信したプローブ要求のシーケンス番号を s_i , RSSI を r_i , 受信した観測機器番号を a_i とする。観測場所に k 個の観測機器があるとして、同時に $T_k \leq k$ 個の観測機器から信号を取り出したいとする。

まず、 $4096 \times k$ のサイズの二次元配列 m を初期化する。4096 はシーケンス番号の最大値で、プローブ要求の並びをこの行列に変換する。またシーケンス番号 i のプローブ要求をいくつ受け取ったか数えておくカウンタ c も初期化する。

次に、全てのプローブ要求について、そのシーケンス番号 s_i と観測機器番号 a_i が

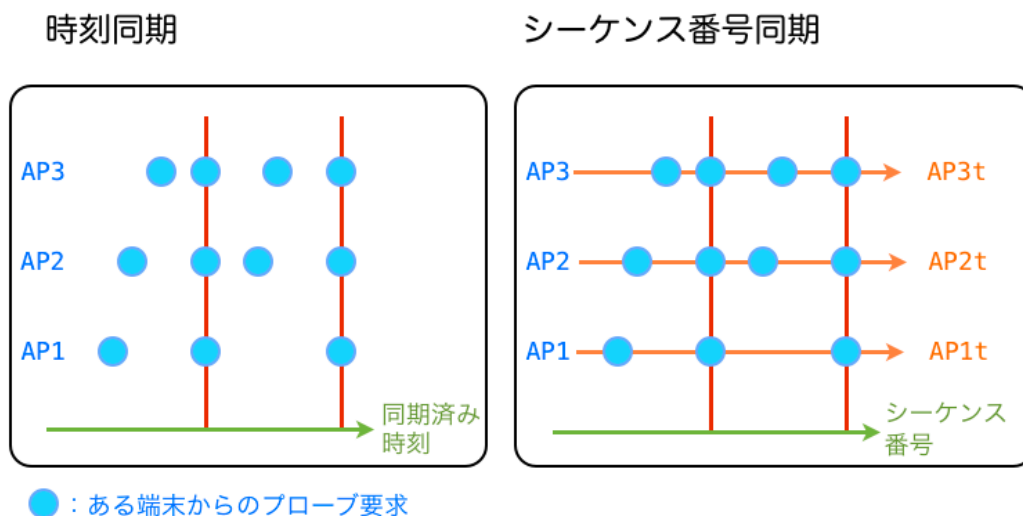


図9 時刻同期とシーケンス番号同期

示す行列の位置に代入していく。ある s_i へ代入されたプローブ要求の数が T_k となったら、行列から RSSI を取り出し処理を行う。

以上のアルゴリズムによってプローブ要求の並びからシーケンス番号でマッチングして RSSI を取り出すことができる。このアルゴリズムの時間計算量は $O(n)$ で、空間計算量は $O(k)$ となる。初期化時に確保しなければならない配列のサイズがやや大きい、この配列はゼロ埋めして使い回しが可能であるため、高速に処理が行える。

このアルゴリズムによって RSSI の組を取り出したら、RSSI の比を特徴量として機械学習を行い、位置推定を行う。 s_i と s_j の比 R_{ij} は、RSSI は対数（デシベル）なため比ではなく差となり、以下のように定義される。

$$R_{ij} = s_i - s_j \quad (3)$$

4.5 端末サンプリング

以上の手法で受信した端末が観測場所内外にあるかを判定できるが、観測機器で受信できる端末は場所内に存在する全ての端末の良いサンプリングとなっているかわからない。各モバイル端末の Wi-Fi 機能がオンになっていないと受信できない上、端末毎にアクティブスキャンの挙動も異なるためである。そのため、受信できる端末は全端末の偏りのあるサンプリングになっていると考えられる。

そのためこのステップで受信した全端末から一部を再サンプリングする。その際の

Algorithm 1 シーケンス番号によるフレームのマッチング

Require: $S = s_0, s_1, \dots, s_n, R = r_0, r_1, \dots, r_n, A = a_0, a_1, \dots, a_k, T_k \leq k$

```
for  $i = 0$  to 4096 do
  for  $j = 0$  to  $k$  do
     $m[i, j] \leftarrow 0$ 
  end for
   $c[i] \leftarrow 0$ 
end for
for  $l = 0$  upto  $n$  do
   $m[s_i, a_i] \leftarrow r_i$ 
   $c[s_i] \leftarrow c[s_i] + 1$ 
  if  $c[s_i] = T_k$  then
    callback( $m[s_i]$ )
  end if
end for
```

サンプリングの基準としては、

- 端末のベンダーによるもの
- プローブ要求の送出間隔によるもの

を用いる。

ベンダーによるサンプリングでは、MAC アドレスの Vendor OUI を用いてベンダーが特定のもの（例えば Sharp や Intel）の端末のみを利用するようにする。ベンダー名はその MAC アドレスが付与された端末の製造業者名を示す。Vendor OUI とは、MAC アドレスの上位 24bit の値で、このビットフィールドと実際のベンダー名が対応している。対応表は IEEE が管理しており [3]、誰でも参照できる。

ベンダーによるサンプリングを用いる理由は、同一ベンダーの製品ではアクティブスキャンの挙動も似ていると考えられるためである。また、ベンダー名からどんな端末か予測できるものもあり、モバイルでないと予想できるものを除くのに役立つ。例えば、ベンダーが Epson の端末は恐らく無線 LAN 内蔵プリンタを示していて、それはモバイル端末ではないので混雑度推定にとってはノイズとなる。

送出間隔によるサンプリングでは、各端末のプローブ要求の送出間隔の中央値を用いて、その中央値が一定値以上の端末のみを利用するようにする。送出間隔が短けれ

ば短いほどその端末の入退室時刻は予測しやすく、長ければ長いほど予測しにくい。そのため、送出間隔が短い端末のみを用いた方が推定結果がよくなると予測できる。

本手法適用の際には、ベンダーと送出間隔の2軸でサンプリングを行い、もっとも良い組み合わせとなるサンプリング手法をヒューリスティックに決定し、その手法を利用する。

4.6 時間補完アルゴリズム

以上で得られたデータから、各端末の入退室時刻を推定、補完する。

4.6.1 入退室制約あり

入退室制約がある場合、その制約を用いて時間補完を行うことができる。観測場所の中にいると判定された端末について、その端末が入退室不可能の時間に退出することはできない。よって、その端末 D_k の ts_k と te_k が入退室可能時間に含まれていない時、その近くの入退室可能時刻まで室内にいたと推定することができる。つまり、 $tc_i \leq ts_k$ の時、 $\max(tc_i)$ を D_k の入室時刻に、 $te_k \leq to_{i+1}$ の時、 $\min(to_{i+1})$ を D_k の退室時刻とする。このアルゴリズムの概念図を図 10 に示す。

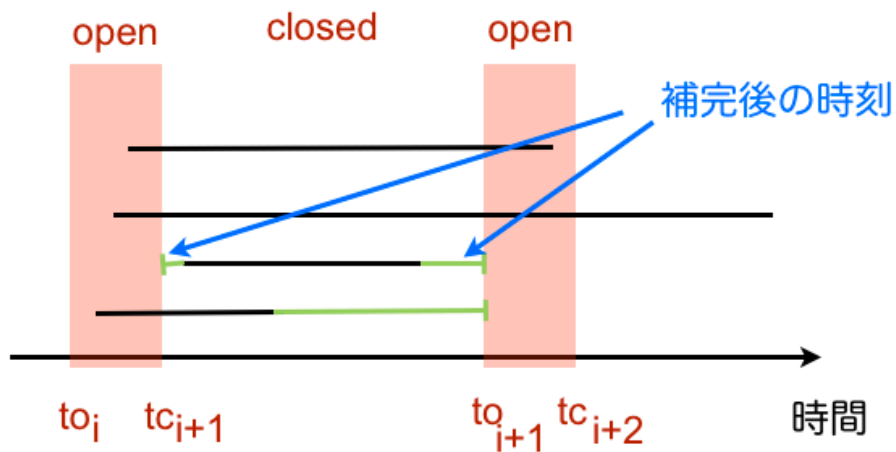


図 10 時刻補完アルゴリズム

4.6.2 入退室制約なし

入退室制約がない場合、端末は自由に観測場所を出入りすることができる。この場合の時間補完アルゴリズムは複数考えられる。

PULSE

最も単純な補完アルゴリズム。位置推定を行い、中にいると判定された時刻のみ在室していて、それ以外の時間は退出していたとする。プローブ要求の送出間隔が計算したい混雑度の時間分解能より小さい端末には非常に有効だと考えられる。

MEDIAN

送出間隔の中央値を用いて補完を行うアルゴリズム。プローブ要求の送出間隔の中央値を m 秒とすると、ある時刻で観測場所内にいると判定されたなら、その時刻の前後 m 秒まで在室していたとする。

ALLDAY

端末をはじめて受信した時刻 ts_k から最後に受信した時刻 te_k まで在室していたとする。端末の途中退室が無い場合に有効なアルゴリズムだと考えられる。

これらのアルゴリズムは端末サンプリングの仕方によって有効なものが変化する。ここのアルゴリズムも混雑度推定結果を見ながらその観測場所に適した物を選択する。

4.7 混雑度推定

以上で混雑度を推定する各時刻 t につき、その時刻 t に端末が観測場所に存在していたかが求まる。ここから、各時刻 t に何個の端末が観測場所に存在していたかを計算する。ここから時系列の端末数データが得られる。この端末数は人数に比例する量になっていると考えられるため、この量を特徴量とし機械学習を行う。求める混雑度が在室人数の場合は単回帰を、質的データの場合はクラス分類を用いる。

混雑度推定の精度を更に向上させたい場合には、時間帯や曜日などの追加の特徴量を用いることができる。しかし、これらの特徴を用いると学習モデルが学習した観測場所に強くフィットしすぎてしまう過学習の問題が生じる。学習モデルの汎用性を重視する場合には端末数のみを特徴量として用いる。

第 5 章 評価実験

提案手法の有効性を確認するため評価実験を行った。実験は電車型の混雑度推定を鉄道車両内で、公園型を大学の教育用計算機室で行った。映画館型は公園型に入退室制約を加えたパターンであり、公園型よりも明らかに簡単なので実験は省略する。

5.1 電車型の混雑度推定

5.1.1 データ収集用システム

電車型の観測場所では観測点が場所と共に移動し、常時観測機器を設置し続けることが難しい。そこで、ノート PC 上で動作させるアプリケーションでデータ収集システムを構築した。アプリケーションは全て Ruby で書かれている。アプリケーションのスクリーンショットを図 11 に示す。

また、推定する目的の混雑度は、鉄道内の人数を自動で収集することはできないため、データを収集しながら目視で混雑度の測定も行う。混雑度は以下の 5 クラスとした。これは NAVITIME の電車混雑レポート [6] を参考にした。

- Level1 [座席が半分以上空いている]
- Level2 [座席に座るスペースがある]
- Level3 [座席が埋まり座れない]
- Level4 [座席前が立つ乗客で埋まる]
- Level5 [ドア前が立つ乗客で埋まる]

また、入退室制約は乗車中に手動で駅到着時刻、発車時刻を記録し利用した。

データは鉄道は西武池袋線, 京王新宿線, 東京メトロ有楽町線の 3 路線で 35 の列車に乗車し収集した。データの一系列は各列車に乗車してから後者するまでとした。

5.1.2 実験結果

収集したデータの一部を図 12, 13 に示す。この 2 つのデータは西武池袋線の 10 両編成の平日 08:57 小手指発準急池袋行 (上り) 列車と平日 17:03 池袋発準急小手指行 (下り) 列車でデータ収集を行ったもので、水平の線分がそれぞれ端末の ts_k, te_k を表している。赤い網掛けの部分は入退室制約で、出入り可能な時間 (駅停車中の時間) を



図 11 電車内用のデータ収集アプリケーション

示している。Station ID と実際の駅名の対応は表 2 に示す。どちらのデータも位置推定を行う前の段階で、700 個程度の端末から受信している。

このデータから位置推定を行い、時間補完を行った結果を図 14, 15 に示す。位置推定後の結果によって列車外の端末が取り除かれ、列車内にいると推定される端末は 60 ～80 個程度であった。

時間補完後の結果から計算した未サンプリングの状態の特徴量（端末数）を図 16,17 に示す。朝の上り列車は徐々に端末数が増え混雑していく様子が、夕方の下り列車は徐々に端末数が減り空いていく様子が確認できる。

次に、35 の列車全てで同様の処理を行い、端末サンプリングの手法を検討した。列車内で受信したベンダーの分布を図 18 に示す。

列車内で受信する端末としては Apple (iPhone, iPad, Macbook) がもっとも多く、

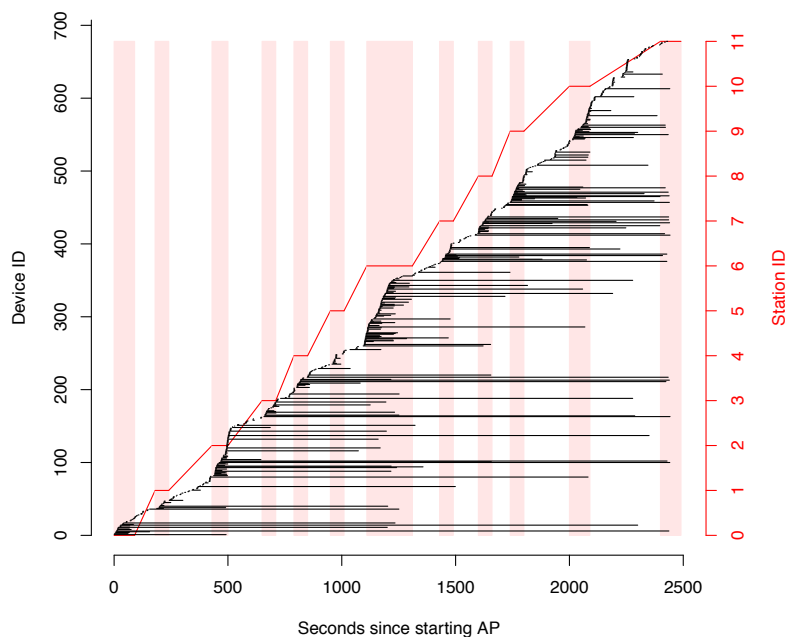


図 12 上り列車で収集したデータ

表 2 Station ID と駅名の対応

Station ID	駅名	Station ID	駅名
0	小手指	6	ひばりヶ丘
1	西所沢	7	保谷
2	所沢	8	大泉学園
3	秋津	9	石神井公園
4	清瀬	10	練馬
5	東久留米	11	池袋

次に Intel (ノート PC)、Sharp (IS シリーズの Android) となった。そこでベンダーによるサンプリングはこの上位 3 つだけの場合と全てを使う場合の 4 種類で行うことにした。平均送出間隔は最小 10 秒から 2 倍ずつ増加させサンプリングを行った。サンプリングの良さの評価値として、混雑度を 1~5 の数値で表して端末数との相関係数を用いた。サンプリング手法の違いによる結果を表 3 に示す。

もっとも良い相関係数が得られたのはベンダーを Apple のみ、送出間隔を 80 秒以下とした場合で、相関係数 0.74 を得た。そのためこの組み合わせを鉄道内のサンプリング手法として採用した。この設定でクラス分類を行った結果の Confusion Matrix

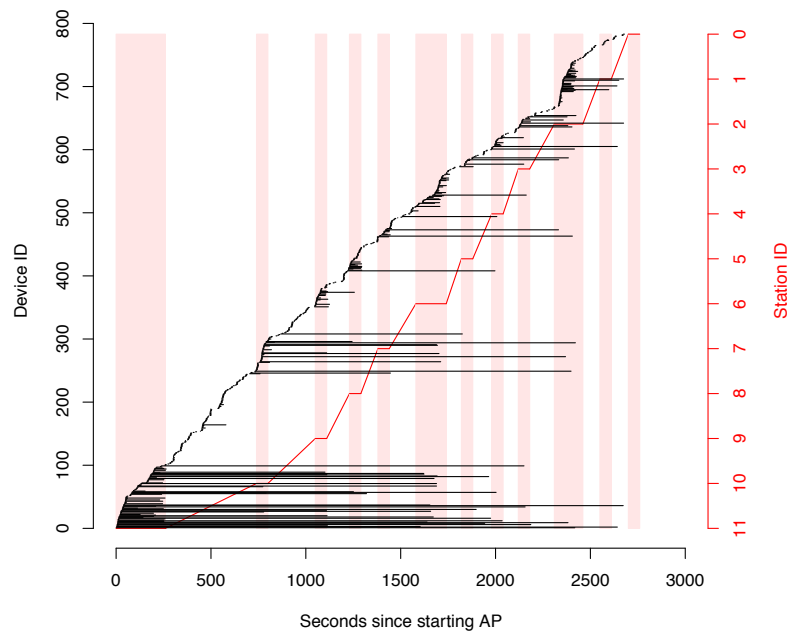


図 13 下り列車で収集したデータ

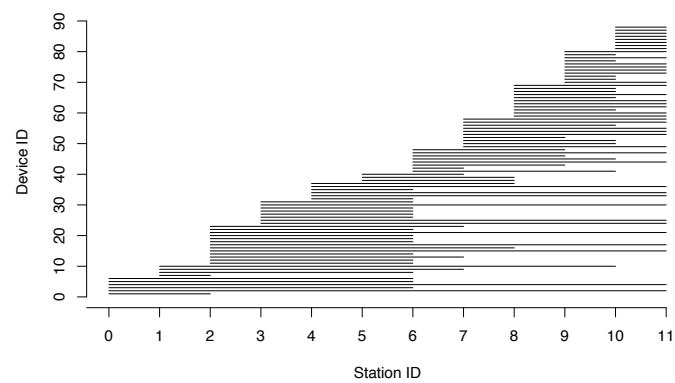


図 14 上り列車での推定乗車区間

を表 4 に示す。全体の分類精度は 74.1% となった。クラス分類器は RBF カーネルの Support Vector Machine を用い、5-Fold Cross Validation で評価を行った。

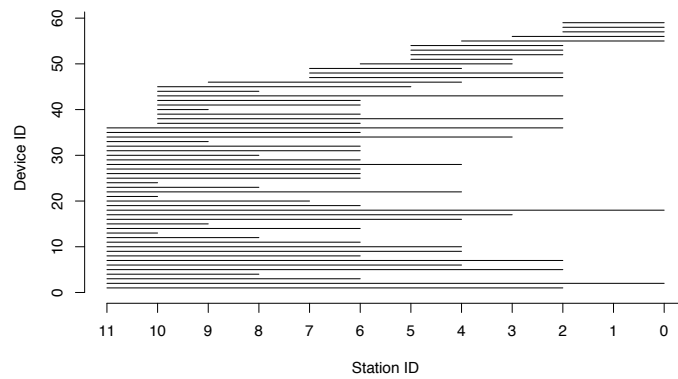


図 15 下り列車での推定乗車区間

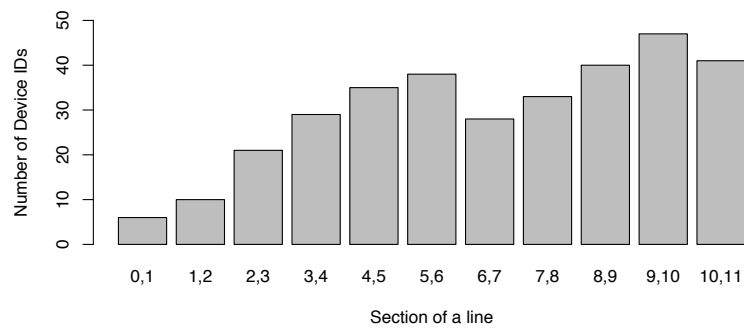


図 16 上り列車の端末数

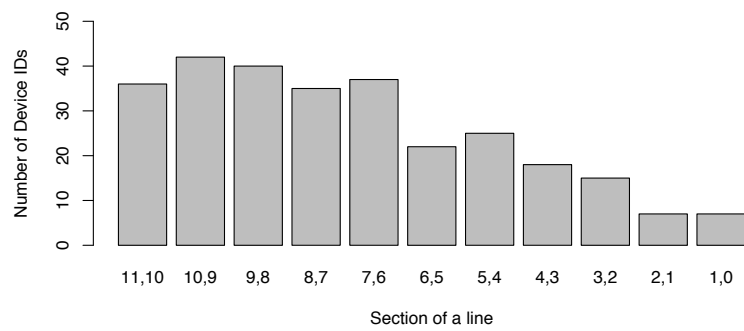


図 17 下り列車の端末数

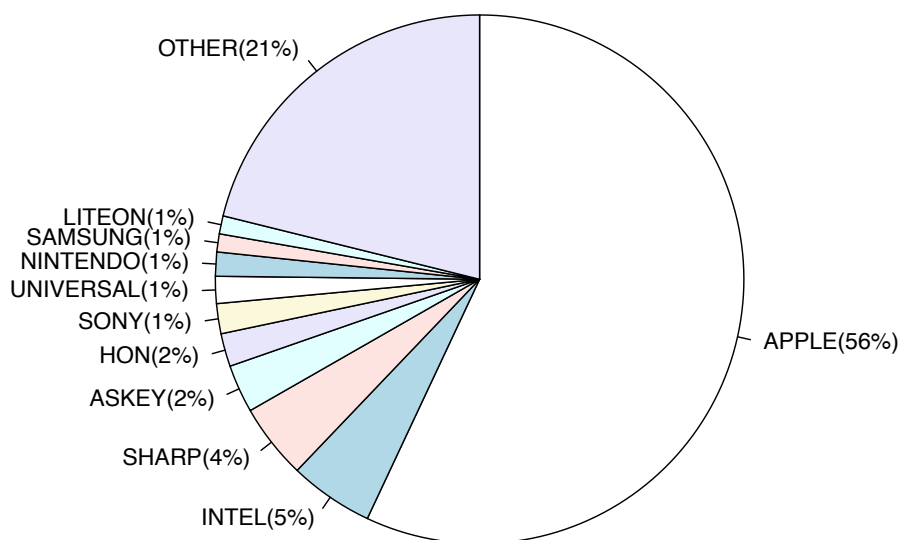


図 18 列車内でのベンダー比率

表 3 端末サンプリングの結果。v=X は、ベンダーが X の端末のみを用いることを、MEDIAN は MEDIAN 秒以下の端末のみを用いていることを示す。

MEDIAN	ALL	v=APPLE	v=INTEL	v=SHARP
10	0.70	0.74	0.06	0.44
20	0.70	0.74	0.04	0.49
40	0.71	0.74	0.05	0.50
80	0.72	0.74	-0.03	0.58
160	0.72	0.73	-0.01	0.58
320	0.71	0.73	-0.01	0.59
640	0.71	0.73	-0.01	0.59
1280	0.72	0.73	-0.01	0.59
2560	0.72	0.73	-0.01	0.58

表 4 混雑度の分類結果

Classified as					
	L1	L2	L3	L4	L5
L1	26	2	0	0	0
L2	18	118	30	6	0
L3	0	4	46	14	2
L4	0	4	4	38	0
L5	0	0	2	2	24

5.2 公園型の混雑度推定

人が自由に出入りする環境が公園型であるので、多くの場所がこの公園型クラスに属する。本論文では大学の建物ほぼ丸ごと一つをカバーするデータ収集システムを構築し実験を行った。

5.2.1 データ収集用システム

電気通信大学の西 9 号館でデータ収集用システムを構築した。システムは 802.11 の規格に習い、プローブ要求をキャプチャするアクセスポイントと、それらを接続するディストリビューションシステムからなる。本論文ではアクセスポイントに無線 LAN ルータを使用する。通常の無線 LAN ルータでは任意の信号キャプチャは行えないが、バッファローのルータはルータ用組み込み Linux ディストリビューションである dd-wrt[5] で動作している。使用されている dd-wrt ではリモートアクセス手段である Telnet や SSH は塞がれているが、ファームウェアアップデート用の TFTP は有効になっている。そこで TFTP 経由で dd-wrt の仲間の Linux ディストリビューションである OpenWrt[4] ^{*1}の独自ビルドファームウェアを流し込み、ルータをリモートアクセス可能な Linux マシン化した。

これらのルータ（アクセスポイント）間を接続するディストリビューションシステムには WDS（Wireless Distribution System）を採用した。WDS は無線によってルータ間を接続する規格で、本来はアクセスポイントのエリアを拡張するために用いられる。WDS を利用することで有線ネットワークを引けないところにもルータを設置することができ、パケットキャプチャの自由度が大きく上がる。しかし WDS へ参加しているルータは、無線媒体が WDS へ束縛されているためパケットキャプチャしてもその WDS 上のトラフィックしか受信できない。そこでルータは 2 台 1 セットで運用することにし、片方を WDS 用、もう片方をパケットキャプチャ用とすることにした。実際に実験に使用したルータを図 19 に示す。

これらのルータを用いて構築したシステムの全体像を図 20 に示す。西 9 号館の 2 階から 8 階までそれぞれ 1 台以上のキャプチャ用ルータが設置しており、電源が 24 時間確保できる場所では常にプローブ要求の信号をキャプチャしている。

システムで受信したパケットはすぐに 7 階にあるサーバへ転送され処理される。しかし階をまたぐ通信は何度もルータをホップする必要があるので、転送には最大 200 ミリ秒程度の遅延が発生する。

^{*1} dd-wrt は OpenWrt からフォークしたプロジェクトで、カーネルの部分はどちらもほぼ同じ。Ubuntu と Debian の関係に似ている。dd-wrt は GUI 指向で OpenWrt は CUI 指向。



図 19 実験に使用したルータ。奥がネットワーク用、手前が信号キャプチャ用

構築したシステムでこれまでに約 10 万個の端末から 2 億のプローブ要求フレームをキャプチャした。

5.2.2 収集したデータ

設置したルータでプローブ要求フレームを受信する度に以下を記録した。

- 受信時刻
- 受信電波強度
- シーケンス制御番号
- 端末の MAC アドレスのハッシュ値
- MAC アドレスから検索したベンダー名
- 受信したルータ番号

受信時刻はルータが受信した時刻ではなく、パケットが転送されてサーバで受信した時の時刻である。そのためネットワーク遅延の 200 ミリ秒程度の誤差がある。

この受信したデータはシステムの DB 内では図 21 の形式で保持されている。

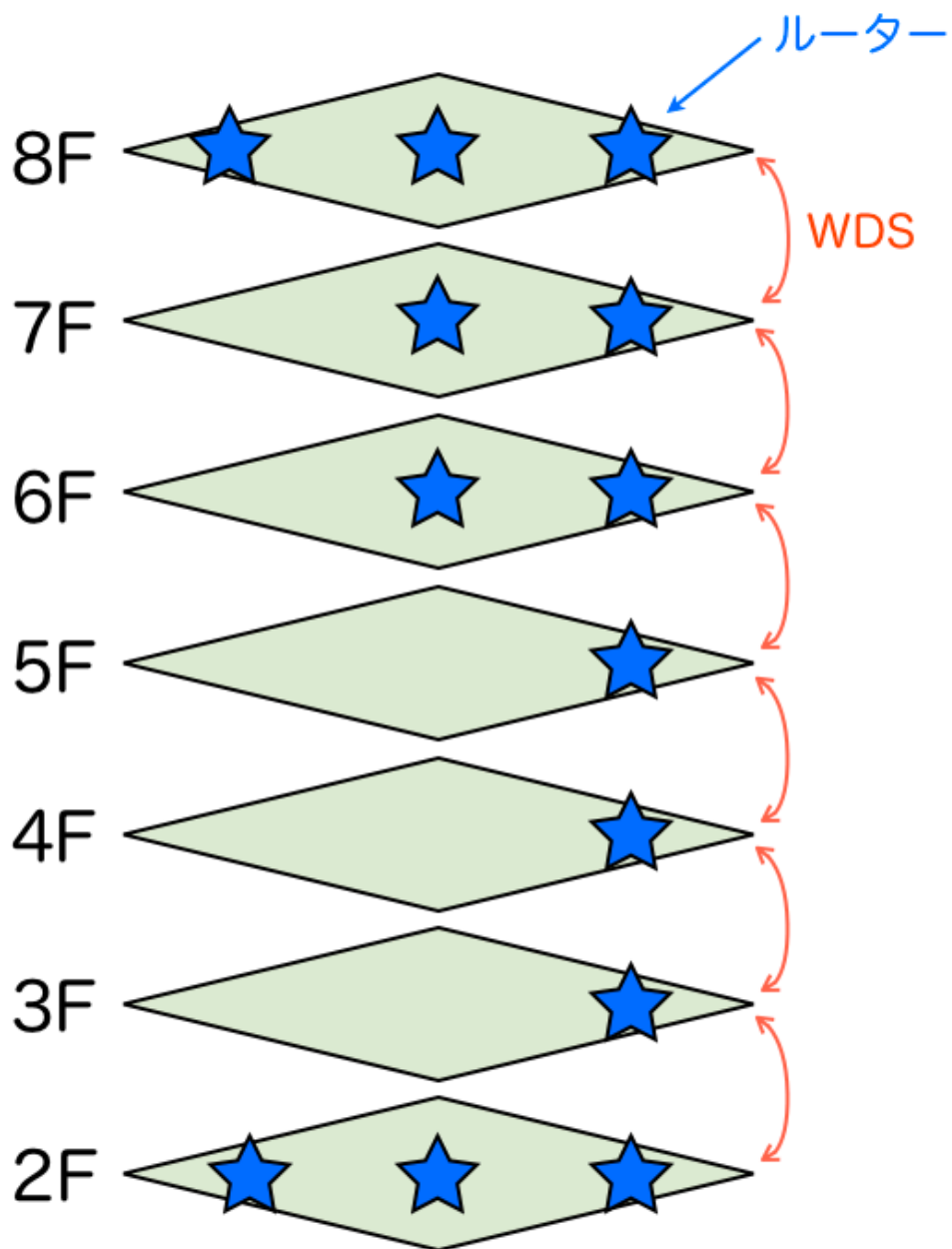


図 20 実験用システム。西 9 号館の 2 階から 8 階までのほぼ全域をカバーしている。

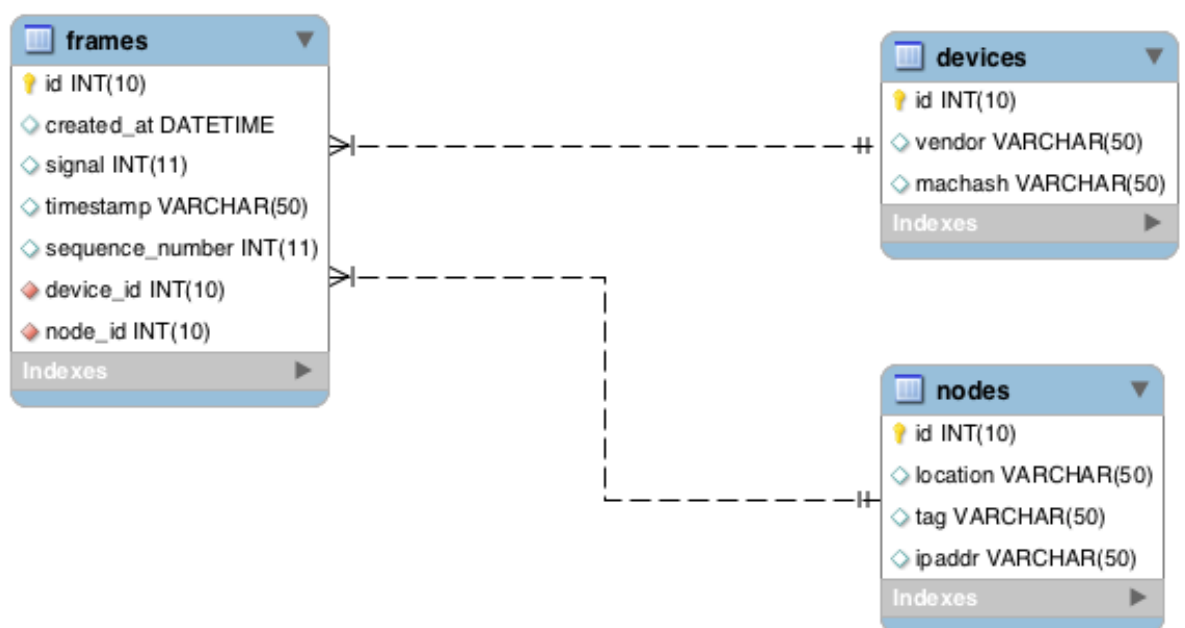


図 21 システムの DB で使用しているスキーマ。frames テーブルがプローブ要求を、devices テーブルがステーションを、nodes テーブルが受信ルータを表している。

また、以下に収集されたデータの実例を示す。

```
{
  "id" : 212374875,
  "timestamp" : 1358064255.5848067,    # サーバでの受信時刻
  "machash" : "8aaf7f77404f468b",      # MAC アドレスのハッシュ値
  "device_id" : 35,                    # ハッシュ値が示す内部 ID
  "location" : "W9-714",               # 受信したルータがある場所
  "signal" : -64,                      # 受信電波強度
  "sequence_number" : 3486,            # シーケンス番号
  "vendor" : "Intel Corporation"       # ベンダー名
}
```

5.2.3 実験用データセット

実験用システムのうち、在室人数の正解データが収集可能な 2 階の教育用計算機室（以下 JED）で実験を行った。教室には約 140 台の端末が置かれており、学生は自由にログインして利用できる。教室の外観とプローブ要求を受信するルータの設置位置を図 22 に示す。実験は 11 月第 3 週に行い、平日の 5 日間（以降 DAY1 ～ DAY5）、9 時から 19 時までの 10 時間データを収集した。この一日分のデータを一列とした。

また、JED の全端末に 1 分毎にリモートログインを行い、ログインユーザを記録した。Linux 端末は `w` (`who`) コマンドでその端末に誰がログインしているか取得することができる。また、ユーザのログイン元も取得できる。ログイン元は、リモートログインの場合は FQDN か IP アドレス、端末に取り付けられたディスプレイからログインしている場合は、`:"0"` になる。よって、例えば `red01` というマシンのリモートログインでないユーザを取得するには以下のコマンドを使えばよい。

```
ssh red01 w -sh | awk '{print $1,$3}' | grep ':0' | awk '{print $1}'
```

これをある時刻に全端末について行えばその時刻のログインユーザ数が求まる。JED は教育用計算機室という性質上、ログインユーザ数は在室人数とほぼ等しいと考えられるため、以降はログインユーザ数を在室人数として扱う。

5.2.4 位置推定の実験結果

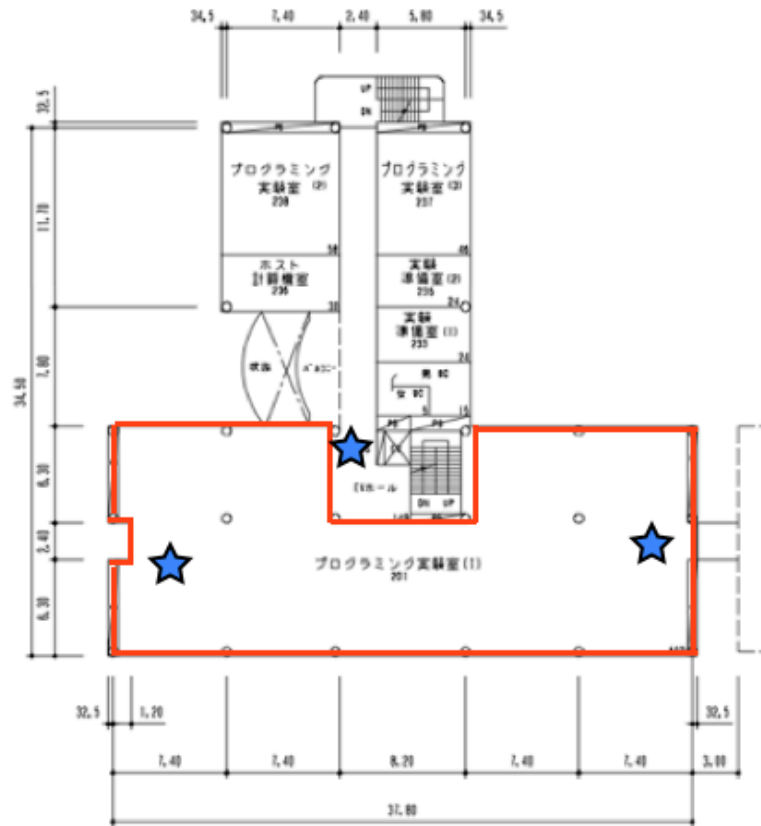


図 22 実験環境の見取り図。赤枠で囲まれた教室が計算機室。星印がルータの設置位置。

以下の 4 つの端末を保持して現在位置を記録しながら JED の内外でアクティブスキャンを行い、データを収集した。

- Apple iPhone 4S (iOS 6.0.1)
- Samsung Nexus S (Android 4.1.1)
- Sony Ericsson Xperia SO-01b (Android 2.3)
- Sharp IS03 (Android 2.1)

受信したプローブ要求から位置推定を行った。RSSI の比にプローブ要求の平均値を使った場合の結果（従来手法 1）を表 5 に、プローブ要求を k-means クラスタリングした結果（従来手法 2）を表 6 に、シーケンス番号でマッチングしてから比を取った結果（提案手法）を表 7 に示す。従来手法 2 の k-means クラスタ数は 5、提案手法の受信アクセスポイント数 T_k は 3 とした。

従来手法 1 の分類精度が 71.1%，従来手法 2 の分類精度が 76.9%，提案手法の分類精度が 76.9% となった。結果から提案手法は従来手法 1 よりも高精度で、従来手法 2 と

表 5 プローブ要求の RSSI の平均値を使った場合の教室の内外の位置推定結果（従来手法 1）

Classified as		
	out	in
out	107	102
in	75	330

表 6 プローブ要求を RSSI でクラスタリングした場合の教室の内外の位置推定結果（従来手法 2）

Classified as		
	out	in
out	121	88
in	56	349

表 7 プローブ要求をシーケンス番号で同期した場合の教室の内外の位置推定結果（提案手法）

Classified as		
	out	in
out	123	86
in	59	346

同程度の精度が得られることがわかる。従来手法 2 とほとんど同じ結果になった理由は、k-means クラスタリングによって抽出されるプローブ要求と、シーケンス番号による同期で抽出したプローブ要求がほとんど同じことによる。このため分類精度はどちらの手法もほぼ同じになったが、計算量は異なる。従来手法 2 ではアクセスポイントの数を a 、アクティブスキャンで送出されるプローブ要求の数を n 、k-means クラスタリングのクラスタ数を k とすると、平均 n/a 個のプローブ要求を a 回のクラスタリングを行わなければならないため、時間計算量は $O(nk)$ となる。一方、提案手法は $O(n)$ で計算できる。そのため提案手法は k-means クラスタリングを用いる手法とは分類精度は変わらないものの、計算量は小さくできるという利点がある。

5.2.5 混雑度推定の実験結果

公園型では入退室制約が利用できないため、混雑度推定するためにはサンプリング手法と補完アルゴリズムの 2 種類を決定しなければならない。そのため、まずベンダーによるサンプリングと補完アルゴリズムとの組み合わせで評価を行った。JED 内

で受信した端末のベンダー分布を図 23 に示す。ベンダーによるサンプリングでは、受信数が多い上位 4 ベンダーをそれぞれ用いる場合と全て用いる場合の 5 種類を用いることにした。サンプリングの良さは端末数と在室人数で評価した。結果を表 8 に示す。

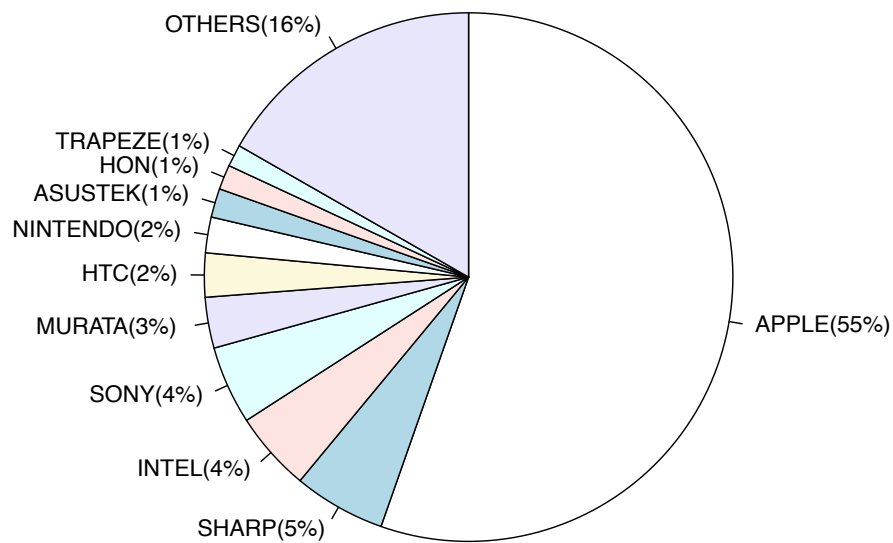


図 23 JED 内のベンダー比率

表 8 ベンダーによるサンプリングと補完アルゴリズムの組み合わせ結果

	ALLDAY	PULSE	MEDIAN
all	0.63	0.67	0.59
APPLE	0.40	0.81	0.73
SHARP	0.10	0.25	0.34
INTEL	0.12	0.37	0.31
SONY	0.10	0.45	0.24

この設定では、ベンダーによるサンプリングで Apple と、補完アルゴリズムに

PULSE を用いた組み合わせが最良の結果となった。

次に補完アルゴリズムはそのままですampling手法に送出間隔を用いた場合の結果を表 9 に示す。

表 9 送出間隔によるsamplingと補完アルゴリズムの組み合わせ結果

MEDIAN	ALLDAY	PULSE	MEDIAN
10	0.10	0.27	0.32
20	0.12	0.30	0.31
40	0.18	0.54	0.54
80	0.26	0.70	0.71
160	0.40	0.68	0.66
320	0.50	0.69	0.67
640	0.57	0.67	0.66
1280	0.63	0.67	0.65
2560	0.65	0.67	0.64

この設定では、送出間隔を 80 秒以下、補完アルゴリズムに MEDIAN を用いた場合が最良となった。次にこれらの結果を組み合わせ、sampling手法をベンダーを Apple、送出間隔を 80 秒としたときに補完アルゴリズムによる違いを表 10 に示す。

表 10 ベンダーを Apple, 送出間隔を 80 秒で補完アルゴリズムを適用した結果

	ALLDAY	PULSE	MEDIAN
v=APPLE & MEDIAN=80	0.83	0.86	0.87

この結果から、ベンダーを Apple、送出間隔を 80 秒で補完アルゴリズムに MEDIAN を用いた場合がこの環境での最良の組み合わせとなる。この組み合わせの場合の各実験日 (DAY1~DAY5) の詳細な結果を図 24, 25, 26, 27, 28 に示す。これらの図は各実験日でどのように在室人数が変化したか (青線)、それに対応する端末数の変化 (赤線) を示している。

また、この設定で全ての実験日に渡って端末数と在室人数をプロットした散布図を図 29 に示す。

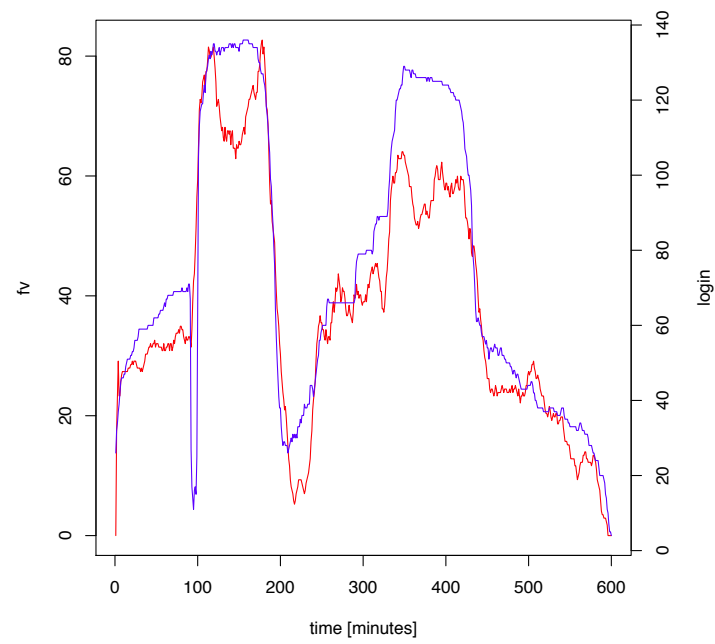


図 24 DAY1 の結果

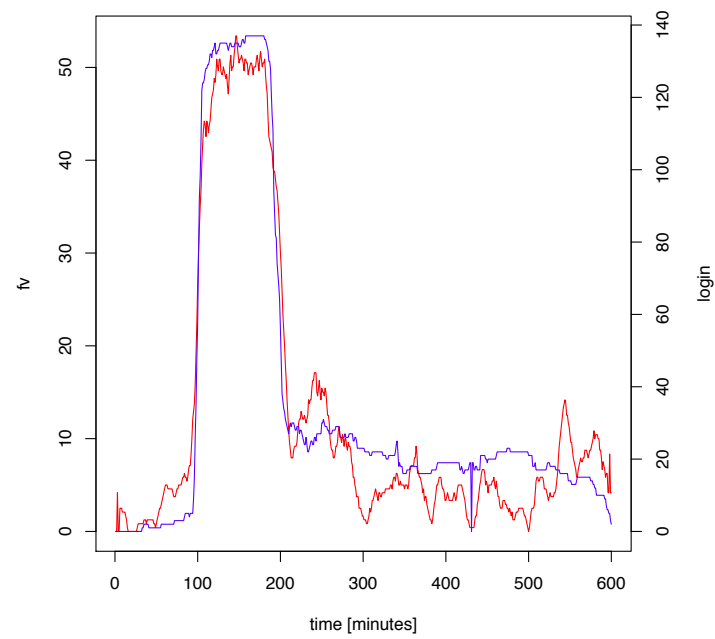


図 25 DAY2 の結果

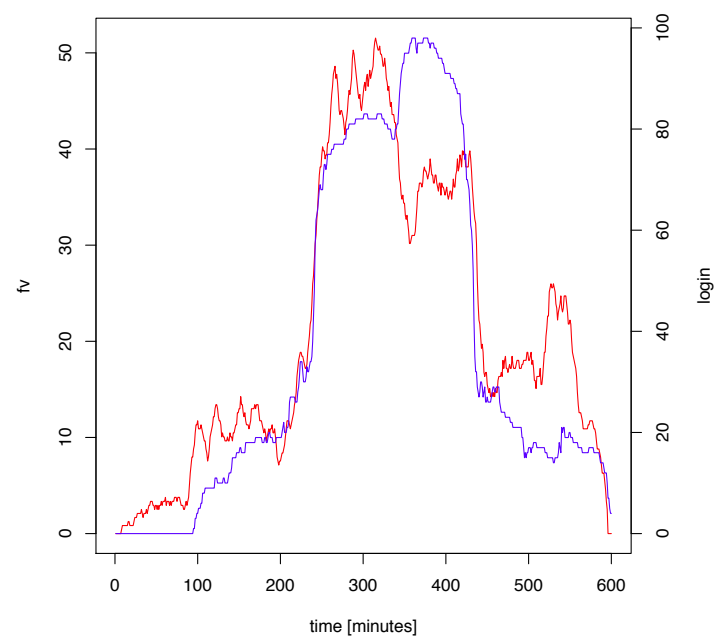


図 26 DAY3 の結果

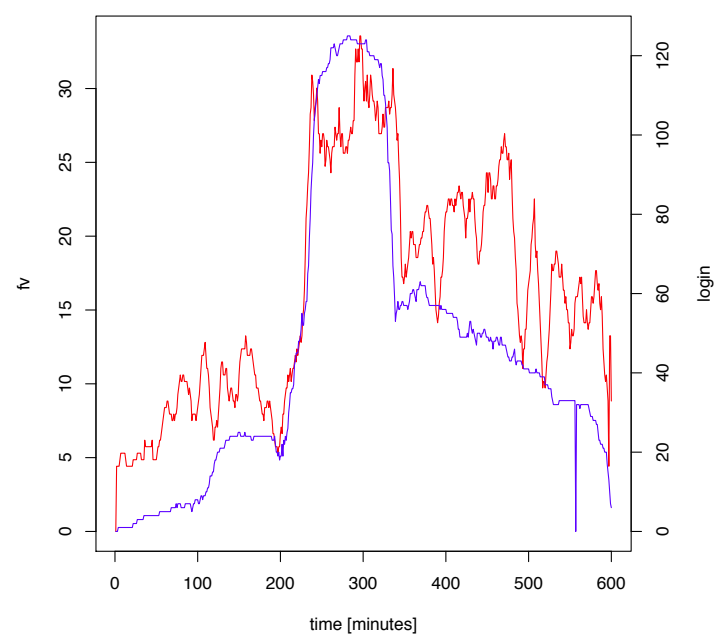


図 27 DAY4 の結果

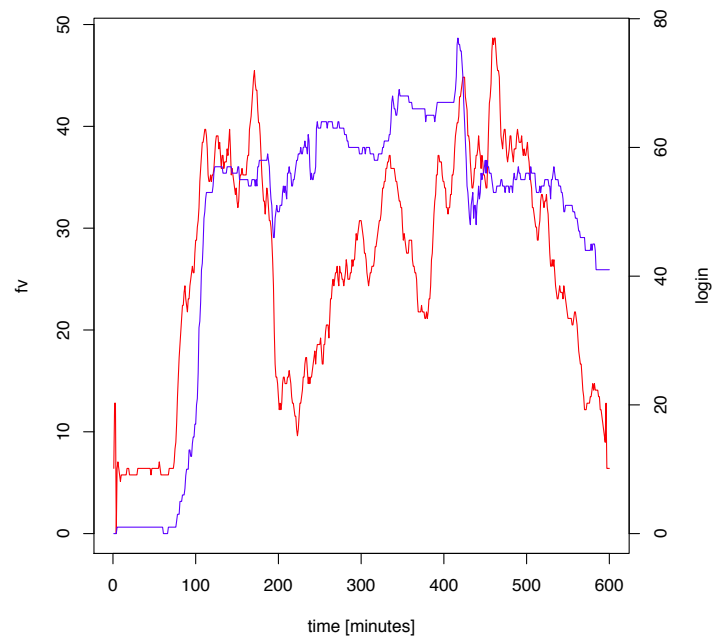


図 28 DAY5 の結果

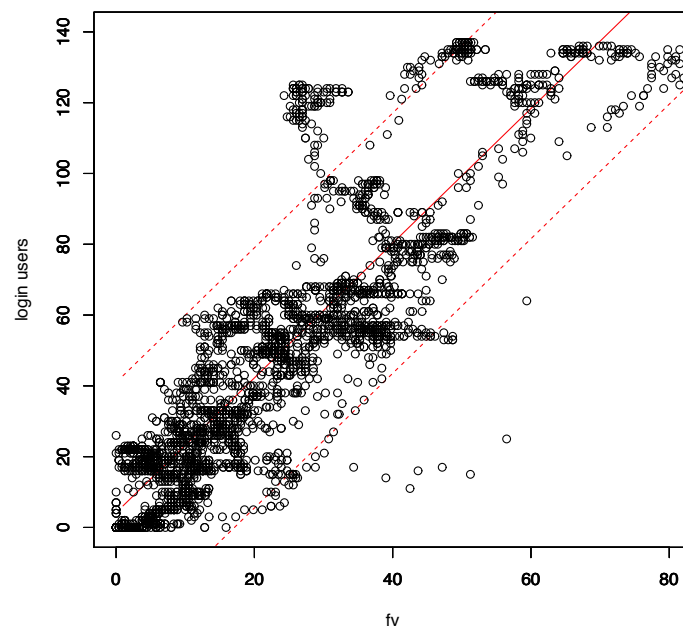


図 29 端末数と在室人数の散布図。赤実線が回帰直線、赤波線が 95% 予測区間。

第 6 章 考察

6.1 端末サンプリング

列車内、教室内ともに最良のサンプリング手法はベンダーを Apple、送出間隔を 80 秒以下と求まった。表 3 から分かる通り、送出間隔によるサンプリングが結果にあまり影響を及ぼさない環境もあるので、送出間隔が 80 秒という基準は普遍的なものとは言えないだろうが、ベンダーを Apple のみとするサンプリングはどの環境でも有効に働くと予想できる。

しかしそもそも端末サンプリングは受信する端末の偏りを減らすために行うものであった。なぜ Apple のみという極めて偏った再サンプリングを行うほうが良い結果となるのだろうか。これは列車内、教室内の端末比率から説明できる。どちらも全体の端末比率（図 18, 23）では Apple が過半数が占めているが、この比率は明らかに観測場所の全端末の一樣なサンプリングにはなっていない。comScore の調査 [7] によれば、日本のスマートフォン市場では Android が 64.1%, iOS が 32.3% のシェアを得ている。観測場所で作られた比率はスマートフォンシェアに反している。ベンダーが Apple の端末はスマートフォンだけではなくタブレットやノート PC も含んでいるが、それでも Apple の比率が高すぎると考えられる。ここから Apple の端末は他のベンダーの端末よりもプローブ要求を送出しやすく、発見されやすい傾向があると言える。つまり観測場所にベンダー X の端末が全部で x 個あったとして、実際はそのうち x' しか発見できない。この時の発見率 $p(X) = x'/x$ とすると、Apple の端末の発見率 $p(\text{Apple})$ は他のベンダーより高くなっているはずである。これによって Apple のみというサンプリングを行うと、その観測場所にある常に発見されやすい端末のみを扱うことができ、混雑度推定結果がより良くなるのだと考えられる。

また、Apple の端末がプローブ要求を送出しやすい傾向は iOS の機能からも説明できる。iOS には無線 LAN を用いて iTunes と同期するための機能 iTunes Wi-Fi 同期がある。この機能を有効化すると、iOS は常に Wi-Fi 接続を確保しようとしてスキャンを繰り返す。それによってプローブ要求が非常に多く送出される。Android にも無線同期（Over The Air update: OTA）があるが、Android の OTA は 3G 回線経由でも実行可能なため積極的に Wi-Fi へ接続しようとはしない。これにより iOS は発見しやすく、Android は発見しにくくなっているのだと考えられる。

6.2 電車型の混雑度推定

全体的に高い判別性能が得られ、混雑度の誤認識もほとんどが1段階の間違いであるから、このような環境において本手法は有効な混雑度推定手法であるといえる。

高い判別性能を達成できた理由の一つに、RSSI を用いなかったことが挙げられる。RSSI は周辺の影響を非常に受けやすい。特に列車内は沿線からの反射波の影響を受けやすい上、常に移動しているので反射波によるノイズの生まれ方が常に変動する。このような環境では従来のノイズ削減手法（前もって反射波も含んだ学習モデルを構築しておき利用する）は適用するのが難しい。しかし本手法では、観測場所の移動と入退室制約を利用することで、RSSI を用いずに位置推定が行えた。

また、高い判別性能を達成できた理由としてもう一つ、鉄道という環境の特殊性が考えられる。首都圏の鉄道の駅付近には各携帯キャリアの Wi-Fi ホットスポットが多く設置されている。キャリアのホットスポットは、接続プロファイルを事前にスマートフォンにインストールしておくことで自動接続するようになっている。従って乗客を乗せた列車が駅に到着すると、接続プロファイルをインストールした端末は一齐にホットスポットに接続しようとする。列車が駅から発車する際には、端末は一齐にホットスポットから切断される。この時にプローブ要求が送出される。つまり、このような端末は駅から発車する度に発見可能である。混雑度推定の時間分解能が駅間であるので、この性質はとても都合が良く、混雑度推定の結果が良かった要因の一つだと考えられる。

本手法の問題点としては、一度観測が終わってから位置推定を行うので、リアルタイムで混雑度推定を行うことはできないことが挙げられる。これは入退室制約が存在する場合、時間補完を行う際に端末 D_k を最後に受信した時刻 te_k が必要になるためである。 D_k からのプローブ要求を受け取った時刻の列を $T = t_0, t_1, \dots, t_n, t_i < t_{i+1}$ とすると、 $te_k = t_n$ となる。時刻 t においてリアルタイム混雑度推定を行うには、 t までは受信したプローブ要求でもっとも最近の時刻 $t_i, \max(t_i) < t$ を用いて時間補完を行う。このときプローブ要求の送出間隔 $t_i - t_{i-1}$ が混雑度推定の時間分解能（例えば列車の場合、前の駅出発から今の駅到着までの時間差）より小さいと、時刻 t における混雑度に D_k を反映できなくなってしまう。この問題は送出間隔による端末サンプリングを強制することで解決できる。端末サンプリングでプローブ要求の送出間隔が時間分解能より小さい端末のみを用いて混雑度推定を行うようにすれば、上記のような状況は発生せず、リアルタイムに推定を行えるようになる。

6.3 公園型の混雑度推定

相関係数は高い結果となったが、部分的に回帰直線にマッチしていない箇所が見られる。これは実験日によってその相関係数が異なることに起因する。実験日単体で見ると、どの日もおおむね良く在室状況を表しているが、全実験日を並べると各実験日によって回帰直線の傾きが少し異なるので、全体としてはばらつきの大きな結果となる。

また、在室人数が多くなるほどばらつきも大きくなっていることが確認できる。このばらつきが生まれた理由は、位置推定手法によるものだと考えられる。本手法では位置推定に RSSI を用いているが、RSSI は人体の影響を受ける。観測場所に多くの人がいるとそれらがノイズとなり、RSSI の値に影響するので、人が多ければ多いほど位置推定精度は低くなってしまう。これを解決するのは非常に難しい。そこで他のアプローチを試みる。本手法では機械学習問題で考えているのだから、計算した端末数に加えて、その端末数がどれだけ信用できるかを示す特徴量が加えられれば精度は向上すると考えられる。そこでアクセスポイントが送出するビーコンを利用する。ビーコンもプローブ要求と同じく人体の影響を受けやすいが、必ず同じ周期で、同じ場所から送出されているという性質がある。これに関連して、人混みが発生するとビーコンの RSSI の分散が大きくなる特徴があるということが阿瀬川ら [15] の研究によって示されている。また進藤 [21] の研究では、RSSI の大きさと人数の間には負の相関関係があることが示唆されている。この性質を利用して、特定のアクセスポイントの RSSI の分散を特徴量に加えることで、位置推定誤差の問題を軽減できると考えられる。

第 7 章 おわりに

7.1 まとめ

無線 LAN 搭載端末はアクティブスキャンを行う度にプローブ要求している。プローブ要求を利用して送出元端末の位置推定が可能である。つまりある地点でプローブ要求をキャプチャすれば周辺にいる端末数を求めることができる。その端末数は周辺人数と比例関係にあると予想されることから、プローブ要求を用いた混雑度推定手法を提案した。鉄道列車内と大学の大教室で実験を行い、有効性を確認した。

7.2 今後の課題

更なる精度向上が今後の課題である。機械学習問題に帰着したことで、本手法を 1 つの特徴量として用い、他の特徴を追加することで精度向上を狙うことができる。他の特徴としては、人混みによって発生する RSSI の影響があげられる。位置推定の分野においては人混みは RSSI に大きなノイズを与えるためそのノイズを削除することが研究課題となっているが、混雑度推定においてはそのノイズそのものが特徴量になりうる。今後はプローブ要求だけでなく、観測場所の周辺の電波状況をより詳細に観測し、混雑度推定に有用な特徴量を見つけていきたい。

謝辞

本研究を行うにあたり、日頃から研究を指導して下さった沼尾雅之教授に深く感謝申し上げます。研究に行き詰まる度に相談にのっていただいた大野将樹助教に深く感謝申し上げます。様々な指摘を頂いた沼尾研のメンバ、特に同じテーマで研究を行った藤井君に感謝申し上げます。

本研究は電通大第 15 回アイデアコンテストに応募し、そこで金賞を頂いたことによりスタートしました。コンテストの会場では様々な意見を頂き、非常にためになると共に、大きなモチベーションになりました。私のブースにお越し下さった皆様に感謝申し上げます。

また、本研究はアイデアコンテストの賞品として電気通信大学ベンチャー支援部門からの資金援助と、研究設備の貸与を受けました。これらの執行にあたり、ベンチャー支援部門の箕輪さんに大変お世話になりました。深く感謝申し上げます。

最後に、この論文を読んで頂いた皆様に感謝を申し上げます。

参考文献

- [1] Core location framework reference. http://developer.apple.com/library/ios/#documentation/CoreLocation/Reference/CoreLocation_Framework/_index.html.
- [2] The google maps geolocation api. <https://developers.google.com/maps/documentation/business/geolocation/>.
- [3] Ieee-sa registration authority. <http://standards.ieee.org/develop/regauth/oui/public.html>.
- [4] Openwrt. <https://openwrt.org/>.
- [5] www.dd-wrt.com — unleash your router. <http://www.dd-wrt.com/>.
- [6] 電車混雑レポート - navitime. <http://www.navitime.co.jp/?ctl=0171>.
- [7] 日本のスマホ利用動向. http://www.comscore.com/jpn/Insights/Press-Releases/2012/8/Japan_Smartphone_Surge.
- [8] P. Bahl and V.N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 2, pp. 775–784. Ieee, 2000.
- [9] M. Gast, 水野忠則, 渡辺尚, 石原進, 峰野博史. 802.11 無線ネットワーク管理. オライリー・ジャパン, 2003.
- [10] M.B. Kjærgaard and C.V. Munk. Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength (concise contribution). In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pp. 110–116. IEEE, 2008.
- [11] J. Krumm and K. Hinckley. The nearme wireless proximity server. *UbiComp 2004: Ubiquitous Computing*, pp. 283–300, 2004.
- [12] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al. Place lab: Device positioning using radio beacons in the wild. *Pervasive Computing*, pp. 301–306, 2005.
- [13] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo. Self-mapping in 802.11 location systems. *UbiComp 2005: Ubiquitous Computing*, pp. 903–903, 2005.

- [14] A Practical Approach to Identifying and Tracking Unauthorized 802.11 Cards and Access Points. Technical report, Interlink Networks, 2002.
- [15] 阿瀬川稔, 田頭茂明, 荒川豊, 中西恒夫, 福田晃. 無線 lan 位置推定における誤差補正手法の一検討 (work in progress, ワイヤレス環境でのアプリケーション品質, p2p/アドホックネットワーク, 画像符号化, ストリーム技術, 信頼性, 一般). 電子情報通信学会技術研究報告. MoMuC, モバイルマルチメディア通信, Vol. 109, No. 122, pp. 13–18, 2009.
- [16] 阿瀬川稔, 田頭茂明, 荒川豊, 福田晃. B-15-11 無線 lan 位置推定におけるプローブ要求のフィルタリング手法 (b-15. モバイルマルチメディア通信, 一般セッション). 電子情報通信学会ソサイエティ大会講演論文集, Vol. 2009, No. 1, p. 501, 2009.
- [17] 伊藤誠悟, 河口信夫. アクセスポイントの選択を考慮したベイズ推定による無線 lan ハイブリット位置推定手法とその応用. 電気学会論文誌 C (電子・情報・システム部門誌), Vol. 126, No. 10, pp. 1212–1220, 2006.
- [18] 伊藤誠悟, 吉田廣志, 河口信夫. locky.jp: 無線 lan を用いた位置情報・測位ポータル. 情報処理学会研究報告, Vol. 2005, No. 90, pp. 25–31, 2005.
- [19] 清水英範. 都市鉄道の混雑率の測定方法. 第 3 回 鉄道整備等基礎調査報告シンポジウム予稿集, 2005 年.
- [20] 荻野敦. 無線 lan による位置検出—日立 airlocation2 (特集: ワイヤレス最前線). 計測技術, Vol. 34, No. 12, pp. 14–18, 2006.
- [21] 進藤和彦. 無線 lan の受信信号強度を用いた迷子防止アプリケーションの検討. 平成 21 年度電子情報通信学会東京支部学生会研究発表会, p. 94, 2010.
- [22] 藤田迪, 梶克彦, 河口信夫. Gaussian mixture model を用いた無線 lan 位置推定手法. 情報処理学会論文誌, Vol. 52, No. 3, pp. 1069–1081, 2011.
- [23] 馬場賢二, 榎原孝明, 湯浅裕一郎. 画像処理による人流計測システム. 東芝レビュー, Vol. 61, No. 12, pp. 35–38, 2006.
- [24] 隆一郎片山, 茂明田頭, 豊荒川, 輝明北須賀, 晃福田. 無線 lan 位置推定システムにおける正準相関分析を用いたデバイス適応手法 (ユビキタス・モバイル, 特集, システム開発論文). 電子情報通信学会論文誌. D, 情報・システム, Vol. 93, No. 10, pp. 1885–1893, oct 2010.
- [25] 中村友宣, 小川剛史, 清川清, 竹村治雄. 二酸化炭素センサによる鉄道車内混雑度推定を用いたウェアラブル学習システムのための利用者コンテキスト認識 (マルチメディア仮想環境基礎, 及び一般, hcg シンポジウム). 電子情報通信学会技術研究報告. MVE, マルチメディア・仮想環境基礎, Vol. 107, No. 554, pp. 49–54, 2008-03-15.

- [26] 暦本純一, 塩野崎敦, 末吉隆彦, 味八木崇. Placeengine: 実世界集合知に基づく
wifi 位置情報基盤. インターネットコンファレンス, pp. 95–104, 2006.