

平成 27 年度 卒業論文

Gradient Boosting Decision Tree を用いた 人間行動認識

平成 28 年 2 月提出

千葉大学 工学部 都市環境システム学科

指導教員：塩田 茂雄

12T0270W 下田 貴久

目次

第 1 章	序論	1
1.1	研究背景	1
1.2	研究目的	1
1.3	論文の構成	2
第 2 章	人間行動認識	3
2.1	人間行動認識とは	3
2.2	人間行動認識に用いる手法	4
2.3	使用するデータセット	4
2.3.1	センサ	4
2.3.2	測定条件	5
2.3.3	前処理	6
2.3.4	データセットのサンプル数	6
2.3.5	特徴量	9
第 3 章	人間行動認識に用いる手法	10
3.1	GBDT の概要	10
3.2	Decision Tree: 決定木	10
3.2.1	CART	11
3.2.2	iris データセットを用いた決定木の例	11
3.3	Boosting: ブースティング	13
3.3.1	Boosting のアルゴリズム	13
3.4	Gradient Boosting Decision Tree	14
3.4.1	目的関数	14
3.4.2	決定木による予測	15
3.4.3	勾配降下法	17
3.4.4	決定木の構築	18

3.4.5	パラメータ	19
3.4.6	パラメータ決定手法	20
3.5	評価手法	20
3.5.1	Multi-Class Log Loss	20
3.5.2	Confusion Matrix	21
3.6	可視化手法	22
3.6.1	PCA: 主成分分析	22
3.6.2	t-SNE	22
第4章	結果	24
4.1	GBDT と SVM の比較	24
4.1.1	パラメータチューニング	24
4.1.2	木の本数と Multi-Class Log Loss の関係	25
4.1.3	Multi-Class Log Loss	25
4.1.4	Confusion Matrix	26
4.2	変数重要度	27
4.3	分類精度と特徴量数の関係	28
4.4	可視化	29
4.4.1	PCA による可視化	29
4.4.2	t-SNE による可視化	29
第5章	結論	32
5.1	まとめ	32
5.2	今後の課題	32
参考文献		33
謝辞		34

第 1 章

序論

1.1 研究背景

近年スマートフォンやウェアラブルデバイスの普及により，位置情報を利用したアプリケーションが増加している．位置情報を取得する方法として最も知られている方法に GPS がある．GPS は，衛星からの電波を使用して位置情報を推定する．このため，電波の届かない地下や電波干渉を受けるオフィス街では思うような精度が得られないことが知られており，こういった場所でも位置情報を取得する研究に注目が集まっている．

屋内で位置情報を取得する方法として Pedestrian Dead Reckoning(PDR) がある (図 1.1)．PDR では，加速度，角速度，磁気センサなどから得られる情報をもとに既知点からユーザーがどれだけ移動したかを推定する手法である．PDR では，前処理としてセンサの値から行動を認識し，行動に応じて，その後の処理を分けることで精度が向上することが知られている [1]．このため，認識精度を向上させることは非常に重要であるといえる．

そこで，本論文では PDR の要素技術である人間行動認識に着目し研究を行った．

1.2 研究目的

既存研究として Support Vector Machine(SVM) を用いて人間行動認識を行った研究がある [2]．本研究では，Gradient Boosting Decision Tree(GBDT)[3][4] を用いて SVM との比較を行った．また，GBDT を用いることにより得られる変数重要度に着目し，使用する特徴量数と分類精度のトレードオフを明らかにした．さらに，PCA と t-SNE[5] を用いて次元圧縮を行い人間行動認識における分類の可視化を行った．

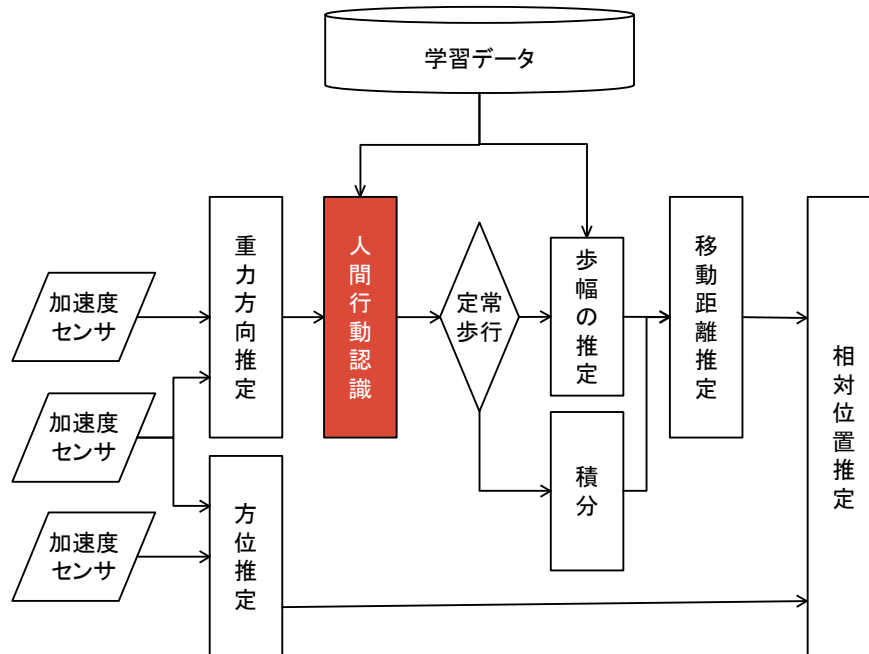


図 1.1 PDR の流れ

1.3 論文の構成

以下に、本論文の構成内容を述べる。

第1章 序論

本研究の概要について述べた。

第2章 人間行動認識

人間行動認識について、使用するデータセットについて説明する。

第3章 人間行動認識に用いる手法

人間行動認識に用いる手法として、本論文で使用する GBDT について説明する。

第4章 結果

GBDT と SVM の比較、変数重要度、分類精度と特徴量数の関係、PCA と t-SNE を用いた可視化についての結果を示す。

第5章 まとめ

本研究のまとめと今後の課題について述べる。

第 2 章

人間行動認識

2.1 人間行動認識とは

一般に，人間行動認識 (Human Activity Recognition: HAR) とは，体に取り付けられたセンサからの情報を利用して，ユーザーが現在行っている行動を認識，判別する問題である．行動認識に使用するセンサとしては，マイクやカメラ，加速度センサ，角速度センサ，方位センサなどが挙げられる．判別する行動は，日常生活でおこなう，歩くや走る，階段を昇り降りするなどの何気ない行動から，スポーツの特定の行動など多岐に渡り，日常生活のモニタリング，スポーツ，医療の分野への応用が期待されている．

従来の人間行動認識では，特別な装置を使用することを前提としたものが多く，日常的に行動分類を行うことができなかった．しかし，近年のスマートフォンやスマートウォッチなどのウェアラブルデバイスの登場により，それらに搭載されたセンサを使用して，より身近に行動認識を行えるようになり，実用化が期待されている．さらに，スマートフォンに搭載されたセンサを用いて分類を行う利点として，日常的に携帯する人が多く，データの取得も簡単で，より多くの人から大量の学習データを集められる点がある．

一方で，従来の特別な装置を使用した人間行動認識とスマートフォンなどのウェアラブルデバイスを使用した人間行動認識を比較すると，ウェアラブルデバイスに搭載されるセンサは，必ずしも高機能なセンサでなかったり，センサの位置が固定されておらず，一定でなかったりするため，精度が低いのが現状である．このため，実際に実用化されているのは，歩数や睡眠時間，活動量 (カロリーの推定) など極めて限られている．

本論文では，より日常的に行動をモニタリングする手法として，スマートフォンに搭載された各種センサを用いた行動認識を扱う．特に屋内位置推定技術として注目されている PDR に適用することを目標として，その場に留まる，座る，歩く，走る，階段を上る，降りるといった簡単な行動を認識することに着目する．

2.2 人間行動認識に用いる手法

人間行動認識を行う手法としては、機械学習を用いた手法が多く取られている。人間行動認識を機械学習の問題として捉える場合、まず、分類対象となる行動とそのときのセンサデータを収集する。次に、ある一定区間(=ウィンドウ)の平均や分散、各軸の相関係数などを特徴量として計算する。こうして得られた特徴量をもとに、それに対応した行動を分類する他クラス分類問題を考え、モデルを構築する。

2.3 使用するデータセット

本論文では、人間行動認識に関するデータセットとして、Human Activity Recognition Using Smartphones Data Set[6]を使用した。

2.3.1 センサ

このデータセットでは、Samsung Galaxy S II に搭載されている加速度センサと角速度センサの値を使用し、横になる (LAYING)、座る (SITTING)、その場に立ったまま留まる (STANDING)、歩く (WALKING)、階段を上る (UPSTAIRS)、階段を下りる (DOWNSTAIRS) の6つの行動に対応したセンサの時系列データを取得している。

図 2.1 に加速度センサと角速度センサの概要を示す。

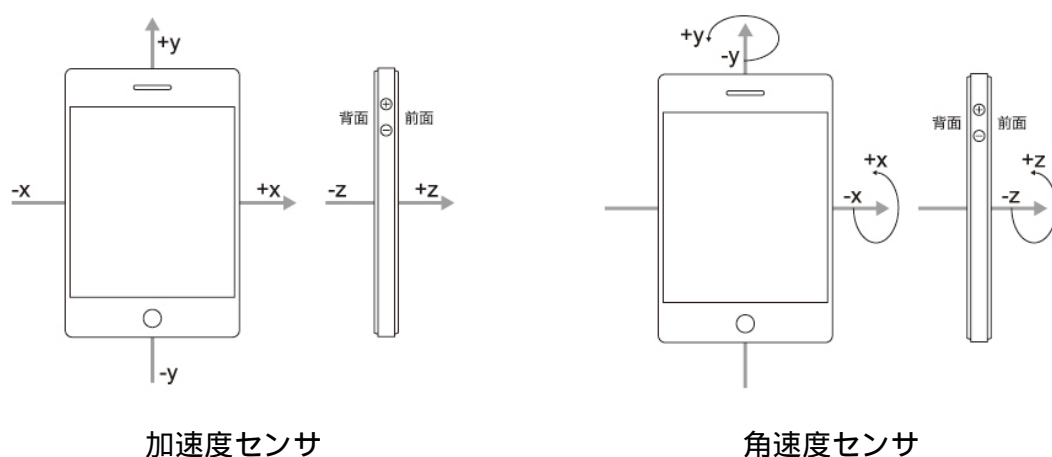


図 2.1 加速度センサと角速度センサ¹

¹ ITmedia Inc. 加速度センサとジャイロで体の動きを感じるアプリを作る
http://www.atmarkit.co.jp/fsmart/articles/ios_sensor02/01.html(最終閲覧日 2016/01/25)

2.3.2 測定条件

データセットのデータは、以下の条件のもとで測定された。

1. 19 歳から 48 歳の男女 30 人のボランティアを対象に実施
2. 分類する行動は横になる (LAYING)、座る (SITTING)、その場に立ったまま留まる (STANDING)、歩く (WAKLING)、階段を上る (UPSTAIRS)、階段を下りる (DOWNSTAIRS) の 6 つの行動
3. Samsung Galaxy S II に搭載されている加速度センサと角速度センサを使用
4. センサの測定頻度 (1 秒間あたりの測定回数) はどちらも 50Hz
5. スマートフォンは、腰の位置に固定
6. 各被験者は 2.1 に示した測定ルートに従い 2 セット測定を行い、合計で 60 回分のデータを収集

表 2.1 測定ルート

No.	行動	時刻 (sec)	累積時間 (sec)
0	スタート	0	0
1	STANDING (1)	15	15
2	SITTING (1)	15	30
3	STANDING (2)	15	45
4	LAYING (1)	15	60
5	SITTING (2)	15	75
6	LAYING (2)	15	90
7	WALKING (1)	15	105
8	WAKLING (2)	15	120
9	DOWNSTAIRS (1)	12	132
10	UPSTAIRS (2)	12	144
11	DOWNSTAIRS (1)	12	156
12	UPSTAIRS (2)	12	168
13	DOWNSTAIRS (3)	12	180
14	UPSTAIRS (3)	12	192
15	ストップ	0	192

2.3.3 前処理

測定されたデータは，そのままでは処理しづらかったり，ノイズが含まれていたりするため前処理を行い，データを扱いやすいように加工する．

データセットでは，まず，Median フィルタと3次元 Low-Pass Butterworth フィルタ(カットオフ周波数=20Hz)を使用してノイズを取り除いている．さらに，加速度センサは，行動により生じる加速度と重力加速度の合力を測定するため，これを行動により生じた加速度と重力加速度に分割している．

こうして得られたデータを一定の区間(=ウィンドウ)に分割している．1つのウィンドウは， $50[\text{Hz}] * 2.56[\text{sec}] = 128[\text{回}]$ としている．これは人間が普通に歩行した場合の1分間の歩数が90[歩/分]から130[歩/分]である[7]に基づいている．さらに，50%のオーバーラップを行い，サンプリングを行っている．オーバーラップとは，連続データであるセンサの値を，ウィンドウのある一定の間隔だけシフトして重複してサンプリングを行うことである．今回の場合，50%のオーバーラップであるので，1.28秒ごとに重複してサンプリングを行っている．

2.3.4 データセットのサンプル数

一般に機械学習では，得られた全サンプルを学習用とテスト用に分割し，学習用のデータを用いてモデルを作成し，検証用のデータを用いて作成したモデルの性能を評価する．データセットでは，前処理を行ったデータに対して，ランダムに選択した70%を学習用データ，残り30%を検証用データとしている．

データセットのサンプル数を表2.2に示す．さらに，前処理後の一つのウィンドウの加速度センサと角速度センサの値を行動ごとに図2.2，2.3，2.4，2.5，2.6，2.7に示す．

表2.2 データセットの各行動のサンプル数

	train	test
LAYING	1407	537
SITTING	1286	491
STANDING	1374	532
WALKING	1226	496
UPSTAIRS	1073	471
DOWNSTAIRS	986	420
SUM	7352	2947

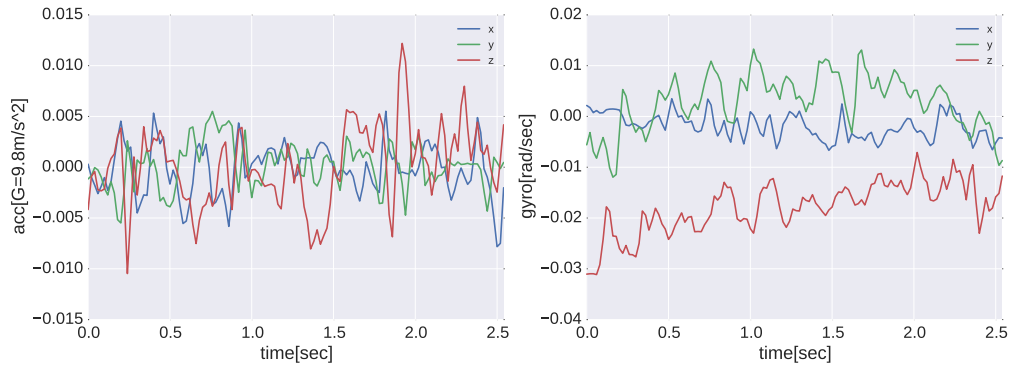


図 2.2 加速度センサと角速度センサの値: LAYING

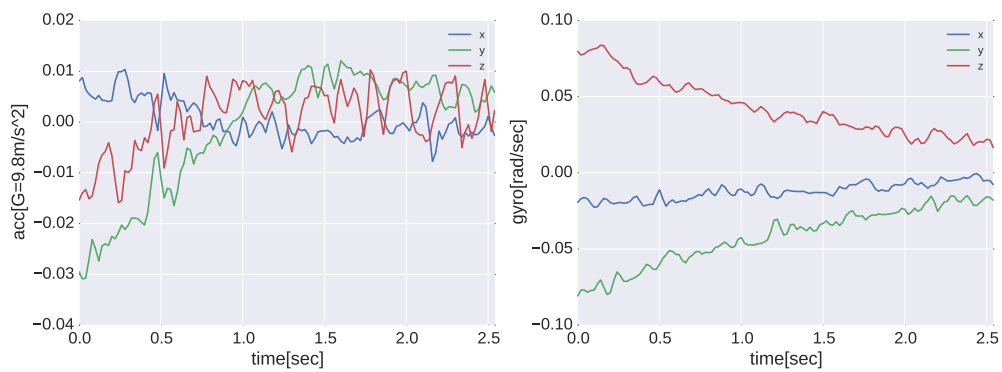


図 2.3 加速度センサと角速度センサの値: SITTING

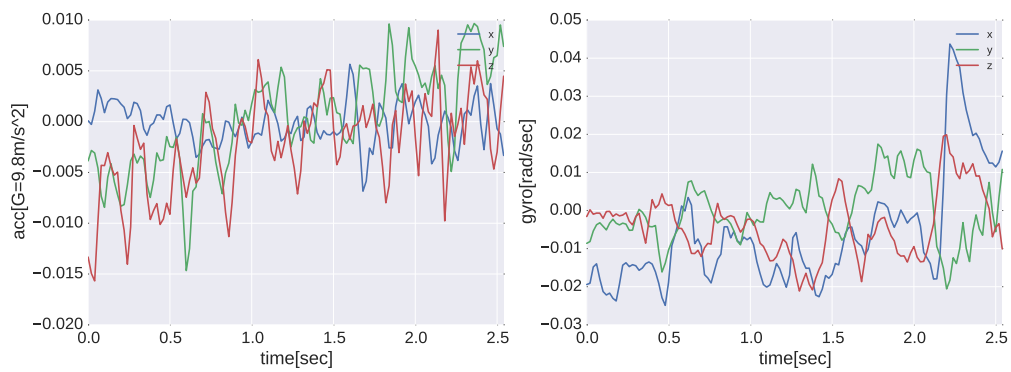


図 2.4 加速度センサと角速度センサの値: STANDING

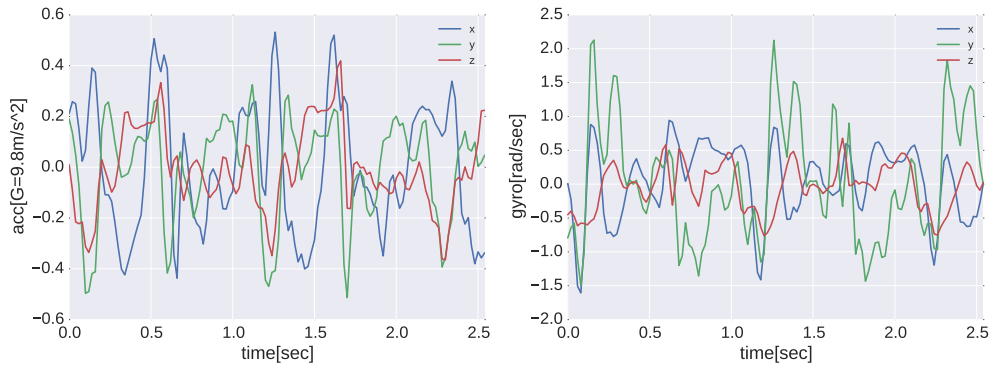


図 2.5 加速度センサと角速度センサの値: WALKING

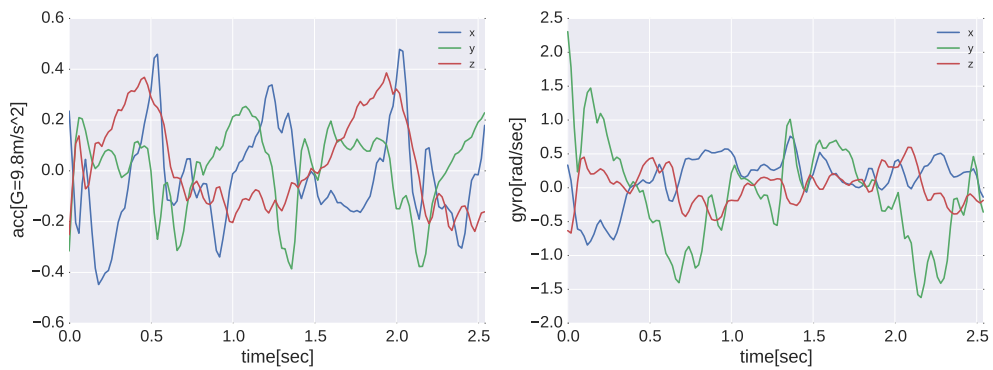


図 2.6 加速度センサと角速度センサの値: UPSTAIRS

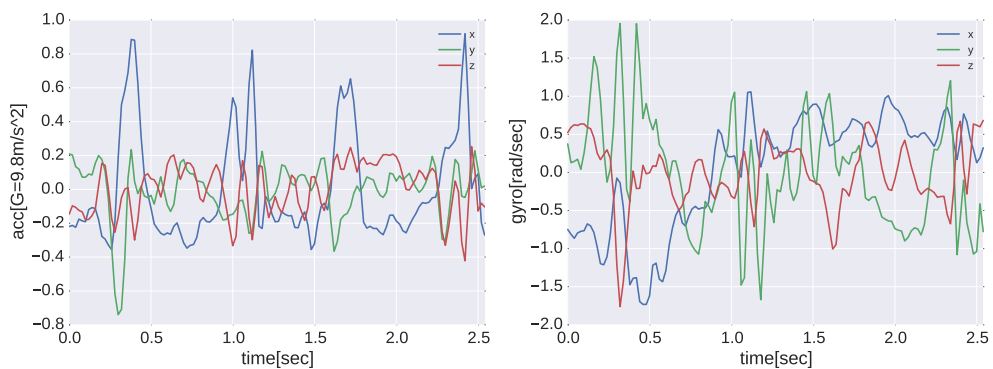


図 2.7 加速度センサと角速度センサの値: DOWNSTAIRS

2.3.5 特徴量

1つのウィンドウには、加速度センサと角速度センサのXYZ軸ごとに128個のデータが含まれる。これらの128個のデータに対して、平均や分散、エネルギー等の関数を適用し、これらの特徴量として使用する。データセットでは、まず、加速度センサの値 $tAcc-XYZ$ を、重力加速度 $tAcc-Gravity-XYZ$ と行動により発生する加速度 $tAcc-Body-XYZ$ に分解している。次に、 $tAcc-Body-XYZ$ と角速度センサの値 $tGyro-XYZ$ に対して躍度 $tAcc-Body-Jerk-XYZ$ 、 $tGyro-Jerk-XYZ$ を計算している。さらに、ベクトルの大きさ $tAcc-Body-Mag$ 、 $tAcc-Gravity-Mag$ 、 $tAcc-Body-Jerk-Mag$ 、 $tGyro-Mag$ 、 $tGyro-Jerk-Mag$ を計算している。さらに、高速フーリエ変換を行ったもの $fAcc-Body-XYZ$ 、 $fAcc-Body-Jerk-XYZ$ 、 $fGyro-XYZ$ 、 $fAcc-Body-Mag$ 、 $fAcc-Body-Jerk-Mag$ 、 $fGyro-Mag$ 、 $fGyro-Jerk-Mag$ を計算している。以上より、得られた33個の変数に対し、表2.3に示す17個の関数を適用して、最終的に561個 ($=33 * 17$) の特徴量を作成している。

表 2.3 特徴量

関数名	説明
mean	平均
std	標準偏差
mad	中央絶対偏差
max	最大値
min	最小値
sma	Signal Magnitude Area $x_{sma} = \sum_{i=1}^n x_i$
energy	エネルギー
iqr	四分位数
entropy	エントロピー
arCoeff	自己回帰係数
correlation	二つのベクトルの相関係数
maxFreqInd	周波数が最大となった位置
meanFreq	周波数の平均
skewness	歪度
kurtosis	尖度
energyBand	周波数間隔のエネルギー
angle	二つのベクトルの角度

第 3 章

人間行動認識に用いる手法

人間行動認識に対して機械学習的アプローチを取る場合、多クラス分類問題となる。これを解く方法として既存研究で用いられている SVM などがあるが、本研究では GBDT を使用し、既存手法との比較を行った。

3.1 GBDT の概要

勾配ブースティング (Gradient Boosting Decision Tree: GBDT) は、決定木を用いて弱学習器を複数構成し、Boosting という手法を用いて逐次的に弱学習器を組み合わせ、最終的に強学習器を作成する手法である。SVM と比較すると GBDT は決定木を用いることで、使用する特徴量が離散的であっても対応が容易で、欠損値や外れ値にも強い学習器を作成することができる利点がある。

以下では、まず、GBDT の構成要素である決定木、Boosting について説明を行い、その後で、GBDT について説明を行う。

3.2 Decision Tree: 決定木

決定木 (Decision Tree) は、データの特徴量を用いた簡単なルールで分岐を作成し、特徴空間を分割することを通じて回帰や分類を行うアルゴリズムのことである。

決定木には、以下の特徴のような特徴がある。

1. 高次元の特徴空間を持った問題でも視覚的に確認することができる
2. カテゴリカルデータに対しても容易に対応できる
3. 欠損値があった場合にも容易に対応できる

3.2.1 CART

決定木を作成する代表的な手法として、CART(Classification And Regression Trees) がある。CART では以下に示す手順で決定木を作成する。

Step.1 決定木の構築

エントロピーや Gini 係数、gain 等の基準を用いて、特徴空間を二分割する手続きを繰り返し行い、決定木を構築する。

Step.2 枝の剪定

Step.1 で構成した決定木の枝のうち重要度の低い分岐を消去する。

Step.2 を行うのは、決定木を使用したアルゴリズムでは、木の深さが深いほど、複雑なデータを扱うことができるが、学習データに対して過度に適合してしまう過学習のおそれがあるためである。

3.2.2 iris データセットを用いた決定木の例

以下では、iris データセットを用いて決定木を構築する例を示す。

iris データセットは、Setosa、Versicolor、Virginic というあやめの品種を、Sepal.Length(がく片の長さ)、Sepal.Width(がく片幅)、Petal.Length(花びらの長さ)、Petal.Width(花びらの幅) という 4 つの特徴量をもとに分類するデータセットである。品種ごとに 50 個のサンプルがあり、合計の 150 個のサンプルがある。

図 3.1 に、iris データセットを用いて決定木を構築した例を以下に示す。図において四角枠で表されるのが、分岐条件であり、丸枠で表されるのが最終的な分類結果である。

また、決定木を構築する際に使用した条件を以下に示す。

表 3.1 条件

条件	値
分岐基準	Gini 係数
木の最大深さ	2

図 3.1 をみると、以下の 2 つの基準に基づいて分類がされているのがわかる。

1. Petal.Width が 0.8cm 以上であるか
2. Petal.Width が 1.75cm 以上であるか

また、決定木の特徴として挙げたように、最終的な結果がシンプルな2分木で表現することができ、視覚的にも理解しやすいモデルを構築することができることがわかる。

図3.1では、よりシンプルなモデルを構築するため、木の最大深さを2に設定したため、それぞれの葉のvalueをみると、Setosaは、全て正しく分類できているのに対し、実際はVirginicaであるサンプルをVersicolorと5つ誤分類し、VersicolorであるサンプルをVirginicaと1つ誤分類してしまっているのがわかる。分類精度を向上させるには、木の最大深さを増加させることで可能であるが、その際は、過学習に気をつける必要がある。

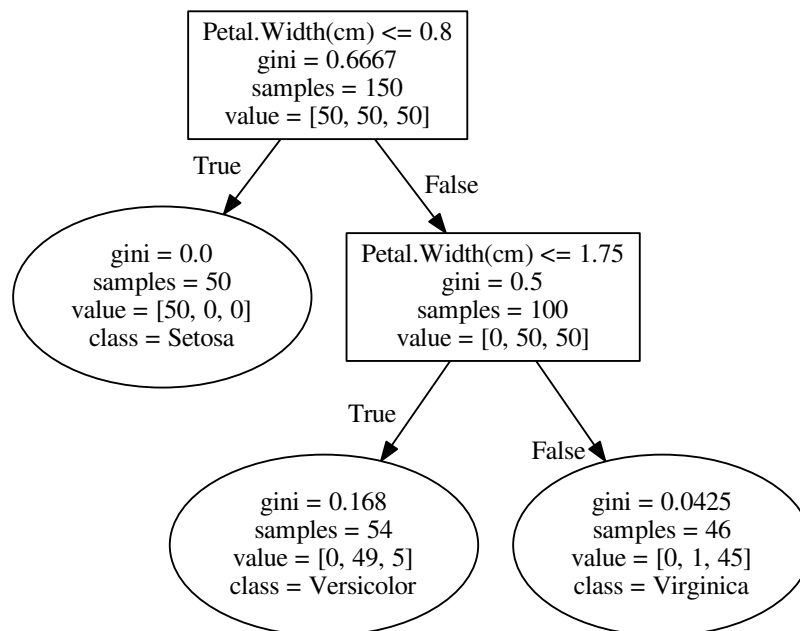


図 3.1 決定木の例

表 3.2 図の説明

項目	説明
gini	そのノードにおける Gini 係数の値
samples	そのノードに属すると予測されたサンプル数
value=[x, y, z]	そのノードにおける実際の Setosa, Versicolor, Virginic のサンプル数

3.3 Boosting: ブースティング

Boosting は、アンサンブル学習 (Ensemble Learning) 手法の一つである。アンサンブル学習とは、多数の弱学習器を組み合わせ、統合することで、より精度の高い強学習器を作成する手法の総称である。Boosting 以外のアンサンブル学習手法としては、Bagging(Bootstrap Aggregating) がある。Bagging が弱学習器を、単純に並列に組み合わせるのに対し、Boosting は、何らかのかたちで、それまでに作成したモデルの結果を反映して新しいモデルを逐一作成する点が異なる。

3.3.1 Boosting のアルゴリズム

特徴空間 $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, ラベル空間 $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$, 訓練集合 $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ を考える。さらに、訓練集合を受け取り仮説 $h: \mathbf{x} \rightarrow y$ を出力する学習アルゴリズム \mathcal{L} を考える。

1. Initialize $D_1 = D$
2. $t = 1$ から $t = T$ まで以下を繰り返す
 - 弱学習器を作成し、仮定 $h_t = \mathcal{L}(D_t)$ を出力
 - 誤差 e_t を計算
 - 新たな訓練集合を作成 $D_{t+1} = \text{CreateNewDistribution}(D_t, e_t)$
3. 弱学習器を統合 $H(\mathbf{x}) = \text{Combine}(h_t(\mathbf{x}))$

GBDT では、学習アルゴリズム \mathcal{L} として 3.2 で示した決定木を使用し、弱学習器の統合には、勾配降下法を使用する。

3.4 Gradient Boosting Decision Tree

3.4.1 目的関数

GBDT では、以下の目的関数を考える。

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta) \quad (3.1)$$

ここで、 $L(\Theta)$ は誤差項、 $\Omega(\Theta)$ は正則化項である。

誤差項 $L(\Theta)$ は、誤差関数 l と、 i 番目のサンプルに対する実際の値 y_i 、 i 番目のサンプルに対する予測値 \hat{y}_i を用いて以下のように定義される。ただし、 N は全サンプル数である。

$$L(\Theta) = \sum_i^N l(y_i, \hat{y}_i)$$

誤差関数 l には、二乗平均誤差 (Mean Square Error) や Log Loss、Multi-Class Log Loss などがあり、問題の種類によって適切に選択する。

以下に、二乗平均誤差の式を示す。

$$l(y_i, \hat{y}_i) = \frac{1}{N} \sum_i^N (y_i - \hat{y}_i)^2 \quad (3.2)$$

二乗平均誤差は、入力データに対して予測値を返すような回帰の問題に対して使用する。以下に、Log Loss の式を示す。

$$l(y_i, \hat{y}_i) = -\frac{1}{N} \sum_i^N (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) \quad (3.3)$$

ここで、 y_i は i 番目のサンプルが真であるとき 1 偽であるとき 0 をとり、 p_i は i 番目のサンプルが真である予測確率であり、 p_i は \hat{y}_i に対応する。Log Loss は、ユーザーはサービスに登録するか、スパムメールであるか等、二項分類であるときに使用する。

以下に、Multi-Class Log Loss の式を示す。

$$mlogloss = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j}) \quad (3.4)$$

ここで、 N はサンプル数、 M はクラス数、 $y_{i,j}$ は i 番目のサンプルがクラス j に属するとき 1 それ以外のとき 0 をとり、 $p_{i,j}$ は i 番目のサンプルがクラス j に属する予測確率で

あり， \hat{y} に対応する．Multi-Class Log Loss は，多クラス分類問題のときに使用する．本論文で扱う人間行動認識は，多クラス分類であるため誤差関数には Multi-Class Log Loss を使用した．

正則化項 $\Omega(\Theta)$ は，機械学習を用いるときにしばしば問題となる，入力データに対して学習し過ぎてしまう過学習を防ぐために導入される．

3.4.2 決定木による予測

GBDT では，複数の決定木によりモデルを構築する．最終的に K 本の決定木を作成したとき， i 番目のデータ x_i に対する予測値 \hat{y}_i は，以下ようになる．

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i)$$

ここで， f_k は， i 番目のデータ x_i に対する k 番目の決定木のスコアを返す関数である．

以下では， t 番目のステップにおいて t 本目の決定木を構築することを考える． i 番目のデータ x_i に対する t 番目のステップでの予測値 $\hat{y}_i^{(t)}$ は以下のように表される．

$$\begin{aligned} \hat{y}_i^{(0)} &= 0 \\ \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ &\dots \\ \hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \end{aligned}$$

ここで，使用する誤差関数を l とすると，誤差項 $L(\Theta)$ は以下の式で表される．

$$L(\Theta) = \sum_{i=1}^N l(y_i, \hat{y}_i^{(t)}) = \sum_{i=1}^N l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) \quad (3.5)$$

また，正則化項は以下の式で表される．

$$\Omega(\Theta) = \sum_{i=1}^t \Omega(f_i) = \Omega(f_t) + constant \quad (3.6)$$

(3.5)(3.6) より , 目的関数 (3.1) を整理すると以下の式で表される .

$$\begin{aligned}
 Obj^{(t)} &= L(\Theta) + \Omega(\Theta) \\
 &= \sum_{i=1}^N l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \\
 &= \sum_{i=1}^N l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant
 \end{aligned}$$

もし , (3.2) に示した二乗平均誤差を使用する場合 , 目的関数は以下のように書くことができる .

$$\begin{aligned}
 Obj^{(t)} &= \sum_{i=1}^n (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^t \Omega(f_i) \\
 &= \sum_{i=1}^n [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant
 \end{aligned}$$

誤差関数

二乗平均誤差を用いた場合では , 目的関数は , 扱いやすく簡単な形にすることができた . しかし , 一般的な誤差関数を考えた場合 , 目的関数が簡単な形になることは少ない . そこで , より一般の誤差関数に対して目的関数を定義するために , 2 階のテイラー展開を行い以下のように目的関数を表す .

$$Obj^{(t)} = \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant \quad (3.7)$$

ここで , g_i は一次導関数 , h_i 二次導関数である .

$$g_i = \partial_{\hat{y}_i^{(t)}} l(y_i, \hat{y}_i^{(t-1)})$$

$$h_i = \partial_{\hat{y}_i^{(t)}}^2 l(y_i, \hat{y}_i^{(t-1)})$$

(3.7) から定数部分を取り除くと , 最終的な目的関数は , 以下のように表すことができる .

$$Obj^{(t)} \approx \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \quad (3.8)$$

正則化関数

正則化関数を導入するにあたり， f_t を以下のように書き換える．

$$f_t(x) = w_{q(x)} \quad (3.9)$$

ここで， q は，データ x を受け取り，そのデータが属する葉のインデックスを返す関数である． f_t は，以下の手順で予測値を返す．

1. データ x を葉 q に割り当てる．
2. 葉 q に対応する重み $w_{q(x)}$ を返す

以上の定義を用いて，正則化関数 $\Omega(f_t)$ を以下のように定義する．

$$\Omega(f_t) = \gamma \mathbf{T} + \frac{1}{2} \lambda \sum_{j=1}^{\mathbf{T}} w_j^2$$

ここで， \mathbf{T} は決定木の葉の数， w_j^2 は j 番目の葉における重みの値， γ, λ は，任意の定数である．

正則化関数を (3.8) に代入すると，目的関数は以下のように表すことができる．

$$Obj^{(t)} \approx \sum_{i=1}^n [g_i f_i(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma \mathbf{T} + \frac{1}{2} \lambda \sum_{j=1}^{\mathbf{T}} w_j^2$$

さらに，決定木では，同じ葉に属するデータは，同じ重み w を共有するので， j 番目の葉に属するデータ x_i のインデックス i の集合 $I_j = \{i | q(x_i) = j\}$ を用いると以下のように書き換えられる．このとき，(3.9) で f_t を再定義したことに注意する．

$$Obj^{(t)} \approx \sum_{j=1}^{\mathbf{T}} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma \mathbf{T} \quad (3.10)$$

3.4.3 勾配降下法

GBDT ではこの目的関数 (3.10) を最適化するにあたり勾配降下法を使用する．

(3.10) 中の $(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2$ は， w_j に関する 2 次方程式となっているため，微分すると以下のように簡単に w_j を求めることができる．

$$w_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

上式より，目的関数 (3.10) は，以下のようになる．

$$Obj^{(t)} = -\frac{1}{2} \sum_{j=1}^T \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma \mathbf{T}$$

以上より，GBDT で使用する目的関数を示せた．

3.4.4 決定木の構築

決定木を構築するにあたり，Gini 係数やエントロピー等の何らかの基準を用いるが，GBDT では，以下のように *gain* を計算し，これに基づいて決定木を逐一構築していく．

$$gain = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} - \gamma \right] \quad (3.11)$$

(3.11) に従って木の最大深さに達するまで茂らせていき，*gain* が負になる枝を剪定する．この作業を指定の回数 (パラメータ：決定木の数) だけ繰り返す．

3.4.5 パラメータ

パラメータについて説明する。GBDT のパラメータは、導入する目的ごとに以下の 2 種類に大別できる。

1. モデルの複雑さを制御するためのパラメータ

- 決定木の数
- 学習率
- 木の最大深さ
- γ (gamma)
- λ (lambda)

2. 過学習を起こさないよう制御するためのパラメータ

- 各決定木で使用するサンプルの割合
- 各決定木で使用する説明変数の割合

決定木の数と学習率

各ステップでの学習度合いを小さくするために、各木の予測値に対して学習率という比率 ν ($0 < \nu < 1$) を掛けた値をモデルの近似式に加算する。 ν を小さくすることで、同じ学習ステップでも最終的なモデルの学習度合いが小さくなる。 ν を小さくする場合、より多くの決定木を構成する必要があり、これらはトレードオフの関係にある。

各決定木で使用するサンプルの割合

各ステップで決定木を構成する際全てのサンプルを使用するのではなく一部 (50% から 100% 程度) のサンプルを抽出して使用する。こうすることで、過学習を起こしづらく最終的に精度の高いモデルを作成できる。

各決定木で使用する説明変数の割合

非常に多くの説明変数を使用する場合やあまり意味のない説明変数が多く含まれる場合などに、この割合を調節する。各決定木で使用するサンプルの割合と同様に、説明変数の一部を使用してモデルを作成することで、過学習を起こしづらく最終的に精度の高いモデルを作成できる。

3.4.6 パラメータ決定手法

3.4.5 で示したように GBDT を使用する場合パラメータを決定する必要がある．本研究ではパラメータ決定手法として一般的な K-Fold Cross Validation を行った．

K-Fold Cross Validation

K-Fold Cross Validation では，与えられるデータを k 個に分割し，検証用データとして使用し，残りの $K-1$ 個を学習用データとして使用する．こうして K 回の検証を行い，得られた K 回の結果を平均してモデルを評価する．これをパラメータを変更ながら繰り返し行い，K-Fold Cross Validation で得られた評価結果が，最も良かったときのパラメータを最終的なテストデータに使用する．

本研究では，5-Fold Cross Validation を用いてパラメータを決定した．図 3.2 に 5-Fold Cross Validation に使用した分割の様子を示す．ここで，白い部分が学習に使用したデータ，黒く塗った部分が評価に使用したデータである．

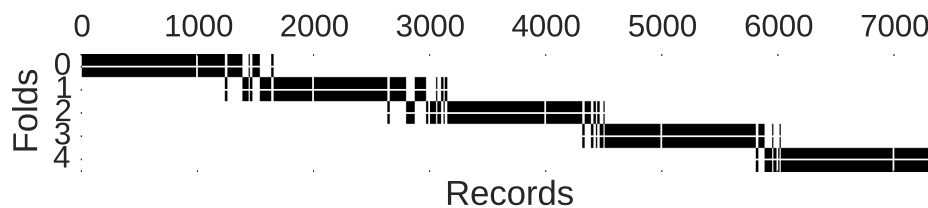


図 3.2 5-Fold Cross Validation

3.5 評価手法

評価手法として 3.5.1 Multi-Class Log Loss と 3.5.2 Confusion Matrix を用いる．

3.5.1 Multi-Class Log Loss

人間行動認識は多クラス分類である．そこで，多クラス分類の評価手法として一般的な Multi-Class Log Loss を使用する．Multi-Class Log Loss は以下の式で表される．

$$mlogloss = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j})$$

ここで， N はサンプル数， M はクラス数， $y_{i,j}$ は i 番目のサンプルがクラス j に属するとき 1 それ以外のとき 0， $p_{i,j}$ は i 番目のサンプルがクラス j に属する予測確率である．

3.5.2 Confusion Matrix

分類を行った結果を評価するとき、真のものを偽と誤って判定する第一種の誤りと、偽を真と誤って判定する第二種の誤りがある。これらを効率的に示す方法として、混同行列 (Confusion Matrix) がある。

TP(True Positive) 真を真と予測

FP(False Positive) 真を偽と予測

TN(True Negative) 偽を偽と予測

FN(False Negative) 偽を真と予測

表 3.3 Confusion Matrix

		予測値	
		Yes	No
実測値	Yes	TP	FN
	No	FP	TN

Accuracy

Accuracy は以下の式で表される。

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

Precision と Recall

Precision と Recall は以下の式で表される。

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

一般に、Precision と Recall は相反する指標である。このため、モデルが最終的にどのように機能するのか、例えば、厳格に誤りを探すのか、それとも、少しでも可能性のあるものをリストアップするのか等、目標に応じてこれらのバランスを調節する必要がある。

3.6 可視化手法

一般に人間行動分類では、非常に多くの特徴量 (今回のデータセットでは 561 個) を用いて分類を行う。このため、2次元平面に各特徴量をプロットすることができず、視覚的に捉えることができない。

そこで、特徴量を2次元に圧縮することで可視化を行うことを考える。こうすることで、2次元平面にプロットすることができるようになり、各行動が、どの程度の距離関係にあるのかを視覚的に捉えることができる。

次元圧縮手法はいくつかあるが、今回は主成分分析 (Principal Component Analysis: PCA) と t-distributed Stochastic Neighbor Embedding (t-SNE) を使用した。

3.6.1 PCA: 主成分分析

主成分分析 (Principal Component Analysis: PCA) は、各説明変数に重みをつけて合成特徴量を作成する手法である。重みは、合成変数ができるだけ多くの元の説明変数の情報を含むように、各説明変数の分散に基づいて決定される。

3.6.2 t-SNE

t-distributed Stochastic Neighbor Embedding (t-SNE) は、もともと高次元データを可視化するように提案された手法である [5]。PCA は広く一般に知られている手法であるが、もともと単に次元を減少させるために提案された手法であり、可視化には向かない場合がある。一方で、t-SNE は、もともと高次元データを可視化するように提案された手法であるため、よりよい良い結果が得られると考えられる。

簡単のため3次元を2次元に圧縮することを考える。t-SNE の特徴として各特徴量間の距離を確率分布で表すことが挙げられる。圧縮前の任意の3つの点 x_i と x_j, x_k に対して x_i を中心とした正規分布を考える。点 x_i と点 x_j の距離は、条件付き確率 $p_{j|i}$ を用いて以下のように表される。

$$p_{j|i} = \frac{\exp \frac{-\|x_i - x_j\|^2}{2\sigma_i}}{\sum_{k \neq i} \exp \frac{-\|x_i - x_k\|^2}{2\sigma_i}}, p_{i|i} = 0$$

また、同時確率 p_{ij} は以下のように表される。

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

次に、圧縮後の任意の3点 y_i と点 y_j, y_k に対しては、正規分布ではなく自由度1のt

分布を考える．t 分布を用いるのは，正規分布と比べると t 分布の方が裾が広いため，離れた値をより遠くに配置することができるためである．点 x_i と点 x_j の距離は，条件付き確率 $q_{j|i}$ を用いて以下のように表される．

$$q_{ij} = \frac{\exp -\|y_i - y_j\|^2 - 1}{\sum_{k \neq i} (1 + \|y_i - y_k\|^2) - 1}$$

このとき，圧縮前の点の距離と圧縮後の点の距離を正確に表現できていると仮定すると， $p_{j|i}$ と $q_{j|i}$ は等しくなる．

ここで，各分布間の距離をカルバック・ライブラー情報量 (KL 情報量) を用いて表す．この KL 情報量を損失関数とする以下の式を最小化するような y_i を求め，それを圧縮後の点とする．

$$C = \sum_i KL(P_i \| Q_i) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

第 4 章

結果

4.1 GBDT と SVM の比較

4.1.1 パラメータチューニング

GBDT は 3.4.5 に示したようにいくつかのパラメータを用いてモデルを調整することができる。同様に SVM も、いくつかのパラメータを用いてモデルを調整することができる。そこで、パラメータ決定手法として代表的な 5-Fold Cross Validation を行い、パラメータチューニングを行った。このとき、評価データに含まれる各行動の割合が各 Fold で等しくなるように調節した。また、パラメータの探索手法には、Grid Search や Random Search があるが、本論文では Random Search を使用した。

各アルゴリズムのパラメータチューニングの結果を、表 4.1 と表 4.2 に示す。

表 4.1 GBDT のパラメータ

パラメータ名	値
学習率	0.125
木の最大深さ	2
gamma	0.01
lambda	1
各木で使用するサンプルの割合	0.5
各木で使用する説明変数の割合	0.5

表 4.2 SVM のパラメータ

パラメータ名	値
C	10
gamma	0.02

4.1.2 木の本数と Multi-Class Log Loss の関係

GBDT の決定木の数と Multi-Class Log Loss の関係を図 4.1 に示す．決定木の本数が 400 を境に誤差がほとんど減少していないことがわかる．このことから，使用するパラメータの決定木の本数を 400 に設定した．

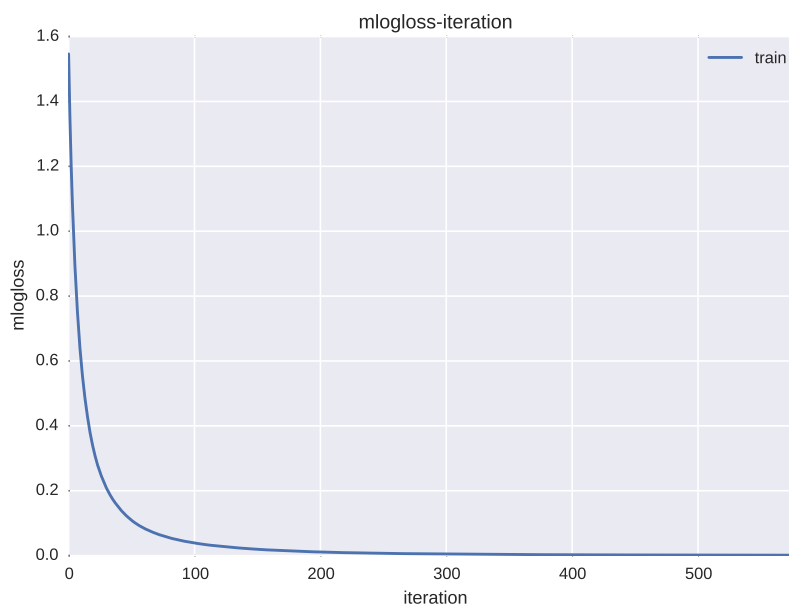


図 4.1 決定木の本数と Multi-Class Log Loss の関係

4.1.3 Multi-Class Log Loss

GBDT と SVM の Multi-Class Log Loss を表 4.3 に示す．GBDT と SVM を比較すると GBDT の方が値が小さくなっていることがわかる．このことから，誤差の観点では GBDT の方が SVM よりもより正確に分類できているといえる．

表 4.3 GBDT と SVM の Multi-Class Log Loss

	GBDT	SVM
mlogloss	0.10484	0.10812

4.1.4 Confusion Matrix

GBDT と SVM の Confusion Matrix を表 4.4 と表 4.5 に示す．両者を比較すると，最終的な分類精度はどちらも 96% であり，分類精度の観点では，両者の間に優位な差が無いことがわかる．GBDT と SVM どちらも LAYING は，100% の確率で分類できていることがわかる．LAYING 以外の行動は，誤分類しやすい行動同士以下のグループに分ける事ができることがわかる．

グループ 1：動的な行動 WALKING，UPSTAIRS，DOWNSTAIRS

グループ 2：静的な行動 SITTING，STANDING

グループ 1 は，各行動により発生するセンサの変化が似ているため誤分類していると考えられる．一方，グループ 2 は，センサの変化が少ないため分類が難しく誤分類していると考えられる．

表 4.4 Confusion Matrix: GBDT

Actual / Predicted	WLK	UP	DWN	SIT	STD	LAY	Recall
WLK	491	4	1	0	0	0	0.99
UP	25	441	5	0	0	0	0.94
DWN	4	16	400	0	0	0	0.97
SIT	0	2	0	444	45	0	0.93
STD	0	0	0	15	517	0	0.95
LAY	0	0	0	0	0	537	1.00
Precision	0.94	0.95	0.99	0.97	0.92	1.00	0.96

表 4.5 Confusion Matrix: SVM

Actual / Predicted	WLK	UP	DWN	SIT	STD	LAY	Recall
WLK	486	6	4	0	0	0	0.98
UP	12	457	2	0	0	0	0.97
DWN	5	25	390	0	0	0	0.93
SIT	0	2	0	446	43	0	0.91
STD	0	0	0	10	522	0	0.98
LAY	0	0	0	0	0	537	1.00
Precision	0.97	0.93	0.98	0.98	0.92	1.00	0.96

4.2 変数重要度

GBDT などの決定木を使用したアルゴリズムでは変数重要度を求めることができる。変数重要度は、各説明変数がモデルの構築にどの程度寄与しているかを示す指標である。図 4.2 に変数重要度が最大の特徴量の値を 100 とした変数重要度の上位 30 個を示す。

図 4.2 をみると、上位に位置する分類により寄与している特徴量は、重力方向に関するもの、行動に伴う X 軸、Y 軸、Z 軸の相関係数に代表されるような各軸間の関係性に関するものであることがわかる。

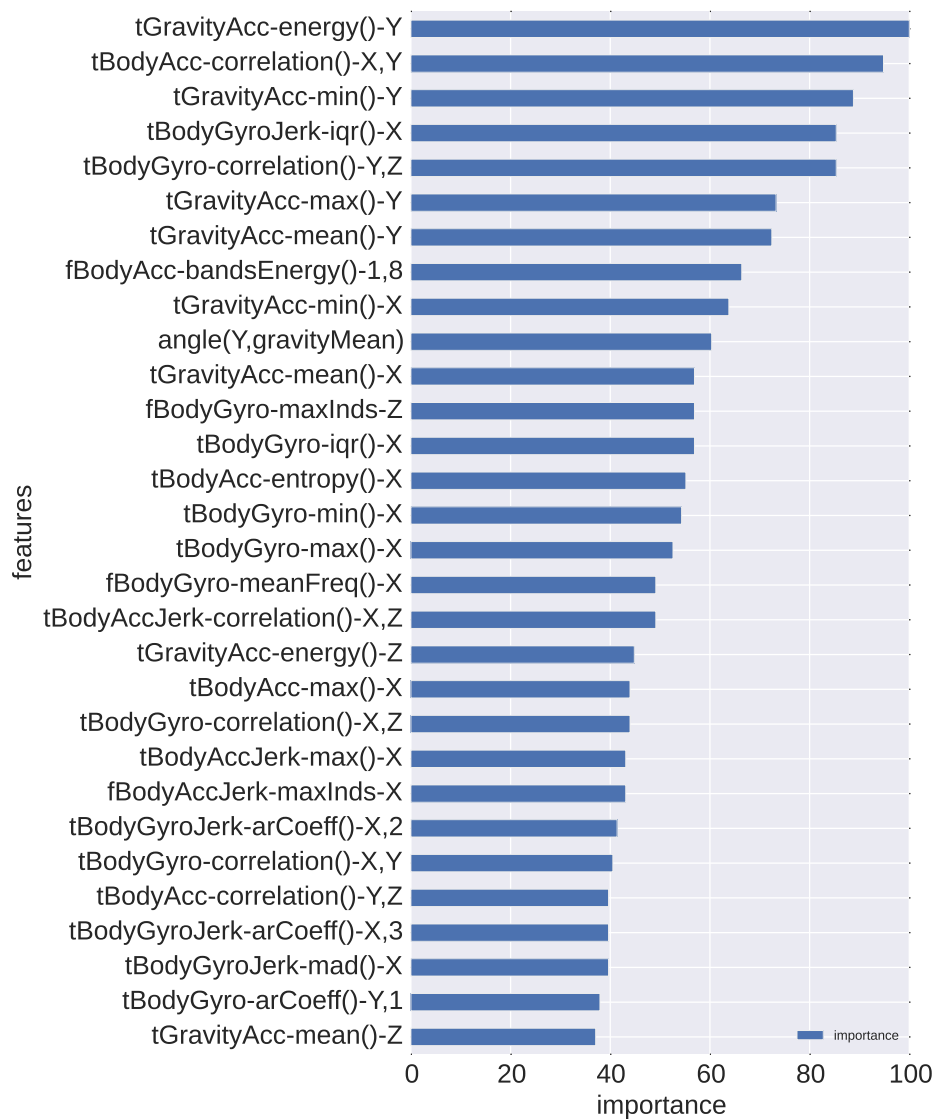


図 4.2 変数重要度 Top30

4.3 分類精度と特徴量数の関係

分類精度と特徴量数の関係を図 4.3 に示す．ここで，右図が Multi-Class Log Loss と特徴量数の関係であり，左図が Accuracy と特徴量数の関係である．また，使用する特徴量は事前に全ての特徴量を使用して学習した結果から変数重要度の高い上位 N 個を選択した．

右図より，評価指標が Multi-Class Log Loss のとき，使用する特徴量が 200 個を境に，ほとんど減少していないことがわかる．同様に，評価指標を Accuracy にした左図でも，使用する特徴量が 200 個を境に，Accuracy が，ほとんど増加していないことがわかる．

以上より，変数重要度が高い特徴量上位 200 個だけで分類を行っても十分な精度が得られることがわかる．また，分類に，ほとんど寄与しない特徴量が，多く含まれていることがわかる．

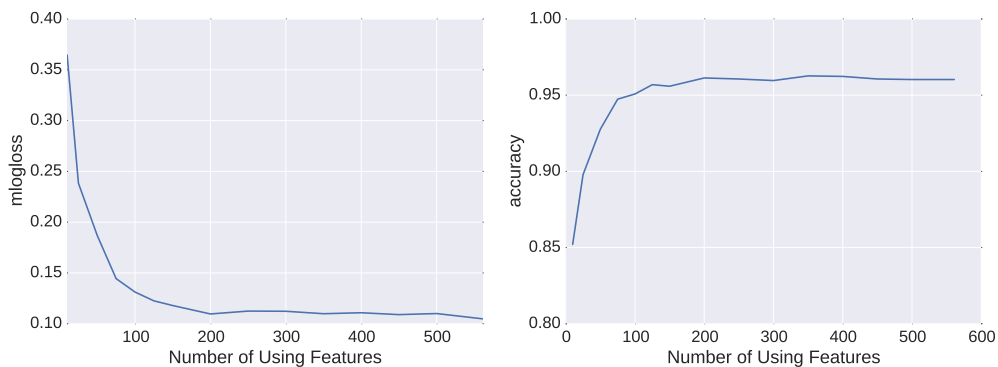


図 4.3 分類精度と特徴量数の関係

4.4 可視化

4.4.1 PCA による可視化

主成分分析 (Principal Component Analysis: PCA) を用いて特徴量を 2 次元に圧縮したとき、実際の値と GBDT の分類結果を可視化したものを図 4.4, 4.5 に示す。

まず、図をみると、PCA による可視化では、以下の 2 つのグループに分かれているのがはっきりとわかる。

グループ 1：動的な行動 WALKING, UPSTAIRS, DOWNSTAIRS

グループ 2：静的な行動 SITTING, STANDING, LAYING

PCA では、各特徴量の分散を基準にするため、図 2.2, 2.3, 2.4, 2.5, 2.6, 2.7 に示したようにセンサの値のレンジが似たもの同士でグループになったと考えられる。

しかし、グループ内の行動に注目すると、グループ 1 では、各行動間の境界が重なる部分が多く、境界が曖昧であることがわかる。また、グループ 2 では、SITTING と STANDING は、上手く分割されているが、LAYING はグループ内全体に分布しており、PCA を用いた可視化では上手く行動が分類できていない事がわかる。

4.4.2 t-SNE による可視化

t-SNE は非常に計算量が非常に多いため、全特徴量を使用して次元圧縮することが出来ない。そこで、まず PCA を用いて 50 次元に圧縮してから、t-SNE を行った。t-SNE を用いて特徴量を 2 次元に圧縮したとき、実際の値と GBDT の分類結果を可視化したものを図 4.6, 4.7 に示す。

PCA を用いた可視化では、各行動が 2 つのグループにはっきりと分かれていたのに対して、図 4.6, 4.7 をみると、t-SNE を用いた可視化では、全体にやや円形状に分布していることがわかる。これは、3.6.2 で説明したように t-SNE が確率分布を仮定して次元圧縮を行うためであると考えられる。

分布を詳しくみると、LAYING などは明らかに行動が分類されており、4.1.4 で示した結果と一致する。一方で、SITTING, STANDING の境界は、曖昧で、うまく分類できていないことがわかる。

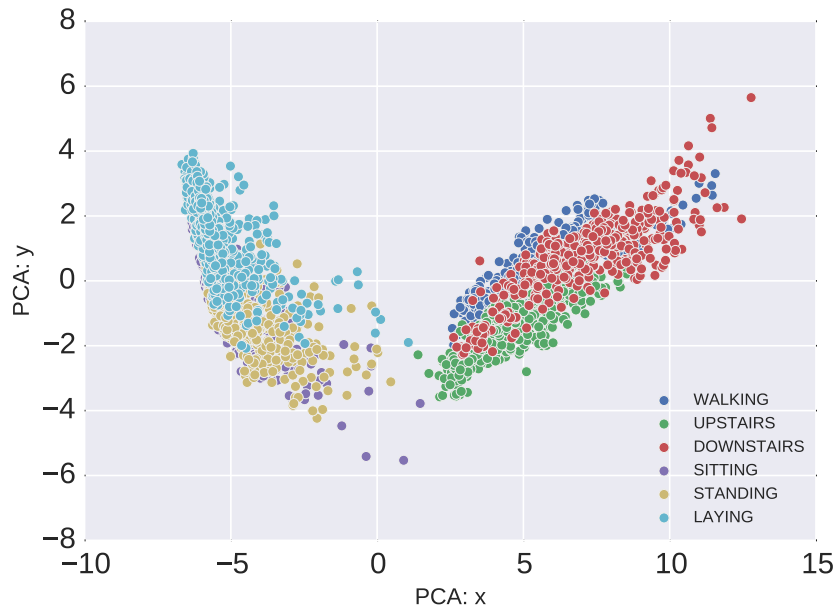


図 4.4 PCA による可視化 (実際)



図 4.5 PCA による可視化 (GBDT の結果)

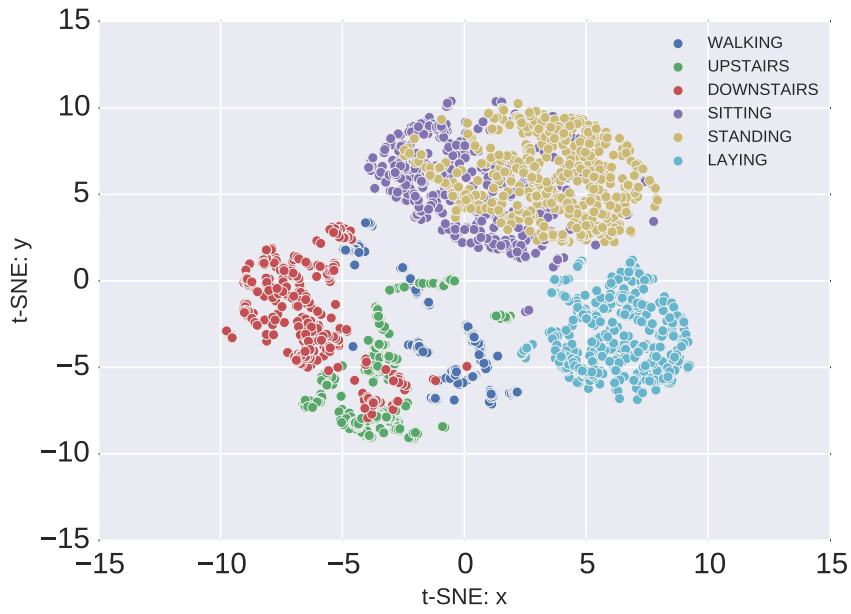


図 4.6 t-SNE による可視化 (実際)



図 4.7 t-SNE による可視化 (GBDT の結果)

第 5 章

結論

5.1 まとめ

本論文では、まず、既存手法である SVM と GBDT の比較を行った。誤差の観点では SVM と GBDT の間に差あるが、最終的な分類精度の観点では両者にほとんど違いがないことを示した。次に、GBDT を用いることで計算することができる変数重要度を用いて、分類に有用な特徴量を特定することで使用する特徴量を削減することができることを示した。さらに、PCA と t-SNE を用いて次元圧縮を行い、人間行動分類の可視化を行った。

5.2 今後の課題

今後の課題として次のようなことが考えられる。まず、今回使用したデータセットは、スマートフォンが腰の位置で固定されていたが、実生活では、スマートフォンを手で持って移動したり、バックに入れていたりすることが多い。このため、今回作成したモデルをそのまま使用することはできない。そこで、加速度センサが、重力方向を検出できることを利用して、前処理を行うことで、スマートフォンが固定されていなくても行動を認識できるようにモデルを改良する必要がある。また、今回のモデルでは、行動の遷移を考慮に入れていない。そこで、マルコフチェインモデルなどを導入して、直前に行った行動も考慮に入れてモデルを構築することが考えられる。さらに、今回の結果を受けて、最終目標である屋内位置推定、PDR への適用を目指す。

参考文献

- [1] 梶航士, 岡田将吾, 新田克己. 歩行者自律測位のための機械学習を用いた動線推定. 2013.
- [2] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Proceedings of the 4th International Conference on Ambient Assisted Living and Home Care, IWAAL'12*, pp. 216–223, Berlin, Heidelberg, 2012. Springer-Verlag.
- [3] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, Vol. 29, pp. 1189–1232, 2000.
- [4] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, Vol. 38, No. 4, pp. 367–378, February 2002.
- [5] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, Vol. 9, pp. 2579–2605, 2008.
- [6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, Bruges, Belgium 2013.
- [7] Chiraz BenAbdelkader, Larry S. Davis, and Ross Cutler. Stride and cadence as a biometric in automatic person identification and verification. pp. 372–377. IEEE Computer Society, 2002.

謝辞

本研究を進めるにあたり，様々なご指導をくださった塩田茂雄教授に，この場を借りて感謝申し上げます．また，多くのアドバイスをくださった塩田研究室の皆様方に感謝します．