Vol.2017-NL-232 No.6
2017/7/19

# シークエンス推定における説明変数の重要度の計算

大北 剛[1]　井上 創造[1]

**概要**：ランダムフォレスト [9] や勾配ブースティング法 [1] において用いられる重要度は簡便なツールで, 分類を終えた後に説明変数の重要さの指標を副産物として得る. 本論文では, このような重要度をシークエンス推定 (シークエンスプレディクション) の設定においても得ることのできる学習法であるアンサンブル RNN を導入する. このアンサンブル RNN は, 分類後に説明変数の重要度を得ることを可能とする. 重要度を計算できるタスクはこれまで分類／回帰タスクであったが, 本論文はこれをシークエンス推定タスクに広げる方法を検討する. なお, 各説明変数は語そのもの (サーフェイス型) では成り立ちづらいため, POS(Part-Of-Speech) タグ/形態素などのクラスによりクラス化したサーフェイス型のクラスを用いる. 深層ディシジョンフォレスト [12] において重要度を得るやり方を応用する.

# Computation of Importance of Random Variables for Sequential Prediction

Tsuyoshi Okita[1]　Sozo Inoue[1]

## 1. イントロダクション

分類タスクにおいて, ランダムフォレスト [9] や勾配ブースティング法 [1] は重要度という簡便なツールを実現する. これらは決定木 [2] の構築法に深くかかわっていたが, 一転して, 分類した後に説明変数の目的変数への貢献度などを求める形で使われている. これまで, シークエンス推定タスクの設定において, (筆者らの知る限り) 重要度という概念を導入したアルゴリズムはない. 一方, アンサンブル学習という側面からはシンプルな RNN 版におけるものは見受けられないものの, エンコーダデコーダー型のニューラル MT においてはアンサンブル学習 [6] が実用的な目的から用いられている. 本論文においては, これらのすべての要素を結合したい. つまり, シークエンス推定タスクの設定において, RNN をアンサンブル学習する形で設定することにより, 説明変数の目的変数への貢献度を求める形で重要度を導入したい. なお, ランダムフォレスト, 勾配ブースティング法などにおいては弱学習器によるアンサンブル学習を使用しているが, RNN においては (強) 学習器である点が異なる. したがって, 深層ディシジョンフォレスト [12] のように弱学習器によりルーチングを確率的に表現する形を考案したわけではない.

## 2. 分類タスクからシークエンス推定タスク

木の構造を $T$,$m$ 番目のターミナルノードを $R_m$, $R_m$ 中の例題数を $n_m$ とすると, $R_m$ においてラベルを $g$ とする確率は $\hat{p}_{m,g} = 1/n_m \sum I[y_i = g]$ となる. また, $R_m$ におけるラベルの予測は, $\hat{y}(m) = \arg\max_g \hat{p}_{m,g}$ となる.

この場合,$T$ におけるノード $m$ のコスト $Q_m(T)$ をジニ係数や情報量ゲイン (エントロピー) で定義できる. ジニ係数の場合 (ジニ係数は CART で導入された [2]), $Q_m(T) = \sum \hat{p}_{m,g}\hat{p}_{m,g'} = \sum \hat{p}_{m,g}(1 - \hat{p}_{m,g'})$ と定義される. 決定木の場合, 分割を行なうことにより、分割後のそれぞれのデータがジニ係数を小さな値にするように分類を続けていくように, 分割ノードを増加させていく形で全体を構成する. Y と N の事象の分岐の確率がある場合に, 2 回試行した場合にイベントが異なるもの YN と NY の合計が、不純さを示す尺度としてのジニ係数を定義する. 一方, 情報量ゲイン [3] の場合, $Q_m(T) = -\sum_{g=1}^{G} \hat{p}_{m,g} \log_2 \hat{p}_{m,g'}$ と定義される.

ランダムフォレストにおいては弱学習器をアンサンブルにするため, 決定木のようにノード分割のアルゴリズムの副産物として得るやり方は使えない. 抜取標本のランダム化というやり方が用いられる。トレーニング集合の説明変数の中で 1 つの変数 $X_k$ に着目する. $X_k$ 以外の説明変数 $X_1, \ldots, X_N$ を固定し, $X_k$ に対応するトレーニング集合の列をシャッフルする. この場合, ジニ係数 (もしくは, 情報量ゲイン) をどれだけ悪くする可能性があるかを評価し, この指数を $X_k$ に対する重要度と考える.

本論文での特殊な設定として, ランダムフォレストと同様にトレーニング集合のランダム変数それぞれに対する重要度を計算することを目的とする. したがって各ランダム変数の条件をさらに細かく分岐する条件を設定することは

[1] 九州工業大学

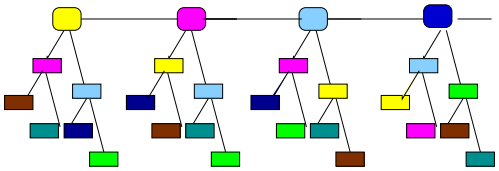

図 1　決定木的なシークエンス推定. もしシークエンス的な要素を考慮しなければ, 上図のように各変数の位置においてその変数を目的変数として他の説明変数から分類を行なう. なお, 上部の列の 4 つの要素が 4 語が文として構成され, それぞれの語からぶら下がった木構造のものが決定木のような役割を果たす. 色付けにも表われるように, 1 番目の語の分類には, 1 番目の語は 1 番目の決定木の内部ノードとはならず, 2 番目の語は番目の決定木の内部ノードとはならない. その他も同様である. なお, われわれの設定において特殊な事情があり, それぞれの説明変数を重要度を計算するため, 説明変数をさらに分解することは考えない. つまり, 図における色分けはそういう意味をもつ.

しない.

## 3. アンサンブル RNN

決定木の場合には応用事例に適用した場合, ランダム変数がどの程度重要であるかはノードの分割順序により明らかとなる. 一方, ランダムフォレスト, 勾配ブースティングなどを応用事例に適用する場合に, 同等の重要度を測定するための方法が重要度 (variable importance) として確立してきた. 以上は分類、回帰タスクにおける話であり, 一般のシークエンス推定タスクを考えると, 学習した後に同様のランダム変数の重要度を求める方法などは確立されていない. 本論文において追求するのは, RNN をアンサンブルにすることにより同等のランダム変数の重要度を求めることができることを示すことである.

### 3.1 アンサンブル RNN のメカニズム

前セクションではシークエンシャルな分類までは行かなかった. この部分を RNN における LSTM の遷移を用いれば, 各々の入力に対するシークエンシャルな依存関係を用いた分類結果としての出力を得ることができる. さらに, 抜取標本のランダム化をこれに追加することで重要度を求める. ただ, これだけだとランダムフォレストのように各学習器を用いることにより得られる抜取標本のランダム化に相当するだけの差違を得ることができない. そこで異なるパラメータを用いて最適化した RNN をアンサンブルとする機構を考案した.

アンサンブル RNN を以下のように定義する. RNN は, 数度トレーニングすることにより, 異なったパラメータを学習することになり, 各々の学習器を弱学習器として重みづけして加算したものを, アンサンブル RNN と呼ぶこととする. このようにして学習したアンサンブル RNN において, それぞれの木のそれぞれの分割における重要度を評価する.

アンサンブル RNN がランダムフォレスト, 勾配ブースティング, 決定木 [9] と異なる点は, 弱学習器のアンサンブルではなく, 通常の学習器のアンサンブルである点である. つまり, RNN やディシジョンフォレストで学習した学習器一つは 1 つのみを独立させて使用することができる. 一方, 弱学習器はアンサンブルで初めて意味をなす.

一つ目の設計基準は, ジニ係数, 抜取標本のランダム化, 情報量ゲインなどの評価尺度のいずれの尺度がアンサンブル RNN に適するかというものである. ジニ係数は勾配ブースティングで用いられ, 抜取標本のランダム化はランダムフォレスト, 情報量ゲインは決定木で用いられている. ディシジョンフォレストにおいては抜取標本のランダム化を用いた場合, 若干, 効果が出難い形となった. 二つ目の設計基準は, 無視される変数が存在するか否かである. 勾配ブースティングにおいては, いくつかの変数を完全に無視する. 一方, ランダムフォレストでは無視される変数は存在しない. 三つ目の設計基準は, 説明変数の重要度に興味があるため, 若干制限された状況にのみ関心がある. 説明変数をオン／オフするような分岐のみを考えることにする.

RNN は以下の式で表現できる. $m$ 番目の RNN モジュールの k 番目の RNN のコンポーネントは通常の RNN と同様に以下のように定義できる.

$k$-th RNN Component

$$\begin{cases} h^{(1)}_{<t>} = \mathrm{LSTM}_{(1)}(h^{(1)}_{<t-1>}, x_t; \tanh) \\ p_k(x_{t,j} = 1 | x_{t-1}, \ldots, x_1) = \frac{\exp(w^{(1)}_j h^{(1)}_{<t>})}{\sum_{j'=1}^{K} \exp(w^{(1)}_{j'} h^{(1)}_{<t>})} \end{cases}$$

これらはすべて弱学習器ではなく, 独立した学習器であるため, エンドツーエンドのアンサンブルとしてのトレーニングは行なわず, 2 段階でトレーニングを行なう. 第 1 段階において, 各々の学習器としてディベロップメント集合 1 を用いてトレーニングを行なう. なお, 各々の RNN コンポーネントは、さまざまなパラメータのものを用意するものとする. 第 2 段階において, 各々の RNN をアンサンブルした結果を学習器としての出力とするため, これをディベロップメント集合 2 を用いて, チューニングを行なう.

$$p(x_{t,j} = 1 | x_{t-1}, \ldots, x_1) = \sum_{k=1}^{M} p_k(x_{t,j} = 1 | x_{t-1}, \ldots, x_1) \frac{\exp(w^{(1)}_j h^{(1)}_{<t>})}{\sum_{j'=1}^{K} \exp(w^{(1)}_{j'} h^{(1)}_{<t>})} \quad (1)$$

### 3.2 アンサンブル RNN におけるアテンションテンソルモデル

POS タグづけタスクにおけるアンサンブル RNN の重要度の役割を考えると, 重要度を求めることはそれ自体 POS タグづけタスク自体の精度に直接貢献しない. しかし, 重要度の高いものに焦点（つまりアテンション）を絞り, こ

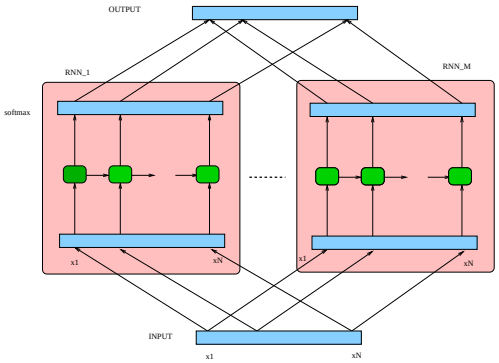


**図 2** アンサンブル RNN の概要．入力が下部にあり，出力が上部にある．中間部に図においては RNN 分類器が 2 つ示されている．これらの RNN 分類器の結果を重み付けしたものを出力結果とする．

れを尊重しながら最適化すれば，全体として重要な呼応関係の正しさを尊重した形で最適解を得られるのではないかと期待できる．POS タグづけタスクのコンテクストにおいて考えると，この重要度は，2 つの単語の照応 (agreement) の起こりやすさを示すものと考えることができる．つまり 2 つの単語の照応をより尊重する形で最適化を行なうという形である [*1]．アテンションをモデルに持ち込むことで全体の精度を向上させるとアイデアは、エンコーダデコーダ翻訳器において，アラインメント行列におけるアラインメントを尊重しながら最適化を行なうことにより飛躍的にトレーニング誤差とテスト誤差が共に上昇したことからの類推である [6][*2]．そこでこのようなアテンションモデルをアンサンブル RNN で実現することを考える．

POS タグづけタスクにおけるアンサンブル RNN を考えると，この設定は通常の設定とは異なり，注目するランダム変数を一つ設定してこれを目的変数とし，その目的変数に対する他のランダム変数の重要度を求めるというように次元が余計に必要となる．各々の $h_i$ により異なるアテンションを用いるため，センテンス長が $L$ の場合，このセンテンスの用いるアテンションは $L \times L \times L$ のテンソルとなる．語の位置 $i(1 \leq i \leq L)$ の語を目的変数とした場合の語以外の説明変数との重要度により，$L \times L$ の行列となり，これが時系列分、つまり L ステップ分の LSTM における時間分だけ時間推移するためテンソルとなる．これ以降このアテンションモデルをアテンションテンソルモデルと呼ぶ．

以上を言い換えると，重要度を POS タグづけタスクにおける「照応」という事象に見たてアテンションと見立てるモデルを考えて、 シークエンス推定を行なう．すると誤差逆伝播のミニバッチにおいて，トレーニング集合全体において照応を高くする特定の単語対をより忠実に結び付けながら最適化を行なう．したがって，この結果できたタスクにおいては，重要度を反映した形で RNN のシークエンス推定をできることになる．

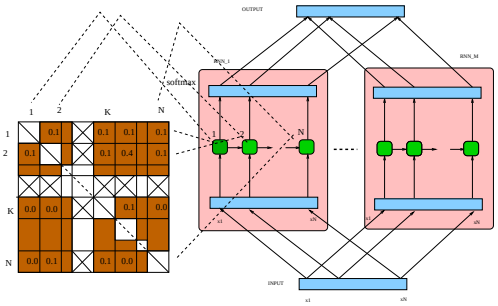


**図 3** 重要度をアテンションとして用いたアンサンブル RNN の概要．アテンションは見易くするためテンソルの 1 スライスのみ表示している．これがセンテンス長さだけ必要なため，テンソルアテンションモデルとなる．

なお，実験において，POS タグづけタスクを取り上げるが，往々にして，シークエンシャルな分類タスクにおいては，語そのもの (サーフェイス型) を説明変数と置くと重要度の計算が非常に細いものとなってしまう．このため，語そのものではなく，語の属する POS/形態素などのクラスによりクラス化したサーフェイス型のクラスを用いる．なお，このようにしても，一般性を失うことにはならない点には考慮されたい．

### 3.3 統計的ばらつきの尺度と良いパラメータの選択

なお，このように定義したアンサンブル RNN において，トレーニング集合のある要素をシャッフルした場合にどの程度、 ジニ係数、 情報量ゲイン、 抜取標本のランダム化の尺度が変化するかを測定し，これをアンサンブル RNN の重要度の定義とする．ジニ係数、 情報量ゲインなどの尺度は，統計的ばらつき (statistical dispersion) の尺度と言われることもある．他に，変動係数 (coefficient of variation)，四分位分散係数 (quartile coefficient of dispersion)，平均絶対偏差 (relative mean difference, Gini 指数の 2 倍)，エントロピーなどがある．抜取 (out-of-bag) 標本のランダム化の尺度は，トレーニング集合においてブートストラップ標本のそれぞれに対して，それ自身が含まれていないブートストラップ標本に対応する木のみの平均を取ったものである．ランダムフォレストにおいては抜取誤差が変化しなくなることでトレーニングの終了とする．

重要度のあるランダム変数とないランダム変数をよりわけるためには，良いパラメータを選択する必要がある [4]．

- サンプル数 $n$ とランダム変数の数 $p$ の感受性 (sensitivity)
- パラメータ $mtry$ と $ntree$ に対する感受性

[*1] 2 つの単語の照応の起こりやすさというテーマは，分散表現を導入した Rumerlhart et al.[8] の論文において登場するテーマである．

[*2] また，統計機械翻訳においては，システムコンビネーション法 [16]，ラティスを使った混合法 [18] など，別々に学習させた機械を結合させる技術が多々ある．そのような場合，単独の機械による性能を遥かに上回る性能を得られることが多い．このようなアーキテクチャを ROVER アーキテクチャと呼ぶこともある [17]．

- ランダム変数のグループとしての重要性が, ランダム変数のグループが極度に依存している場合に依存していない変数を正しく認識できるかという問題

依存するランダム変数に対して重要度を過剰に与えるという問題がある [5]. パーミュテーションテスト (permutation test) は, あるランダム変数が他のランダム変数から独立であるかをテストするものだが, ランダム変数が独立ではなく依存する場合, パーミュテーションテスト (permutation test) が不良設定 (ill-posed) となる. このことにより, 依存するランダム変数に対してより大きな重要度を与えてしまうことになる.

## 4. 実験結果

データは CoNLL2000[14] で用いられた POS タグデータを借用して POS づけタスクを用いた. このデータは本来チャンクデータに対するデータをトレーニング/テストする目的で準備されたデータであり, Penn ツリーバンク [*3] をトレーニング集合として構築した Brill の POS tagger[15] でタグづけされたものである. つまり, 人が介在してアノテーションを行なったものではないため, POS タグタスクの評価としては良いものではない. しかし, 暫定的な版ということで本システムの評価としたい.

アンサンブル RNN, テンソルアテンションモデルにおいては, パラメータの異なる版を 20 個用意した. Lasagne[*4]/Theano[13] で実装を行なった.

RNN 単体としての性能がすでに高いことから, また, アンサンブル RNN においてパラメータの異なる版では差違が若干少ないことも影響しているものと思われ, 性能の改善はそれぞれ 1.1%, 1.6%と小さい.

| | accuracy | unk accuracy |
|---|---|---|
| RNN 単体 | 0.931 | 0.860 |
| アンサンブル RNN | 0.942 | 0.860 |
| テンソルアテンションモデル | 0.946 | 0.860 |

**表 1** 暫定的な実験結果を示す.

## 5. 結論

本論文においては, シークエンス推定タスクの設定において, 重要度を測定できるアンサンブル RNN を提案した. 評価においては, エクストリンジックな暫定的な評価として, この機構を用いたアテンションモデルの性能を測定した. 1.6%の改善を行なえた.

今回の実験においては, テストセットの精度がすでに高い対象を選んだため効果がわかりにくかった. テストセットの精度が中程度な対象を選び, 効果を確かめたい. また,

[*3] Treebank-3. カタログ番号 LDC99T42 で以下で手に入る `https://catalog.ldc.upenn.edu/ldc99t42`

[*4] URL は `https://github.com/Lasagne/Lasagne` である.

### 参考文献

[1] Mason, L.; Baxter, J.; Bartlett, P. L.; Frean, Marcus. ”Boosting Algorithms as Gradient Descent”, In S.A. Solla and T.K. Leen and K. Mller. Advances in Neural Information Processing Systems 12. MIT Press. pp. 512518, 1999.

[2] Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. “Classification and regression trees”. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. 1984.

[3] Quilan, J.R. “Induction of Decision Trees”. Machine Learning 1: 81-106, Kluwer Academic Publishers. 1986.

[4] Robin Genuer, Jean-Michel Poggi, Christine Tuleau-Malot. “Variable selection using Random Forests”, hal-0075548, 2012.

[5] Carolin StroblEmail author, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin and Achim Zeileis, “Conditional variable importance for random forests”, BMC Bioinformatics20089:307, 2008.

[6] Kyunghyun Cho, Bart van Merrienboer Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, Yoshua Bengio, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, EMNLP 2014.

[7] Seep Hochreiter and Jurgen Schmidhuber, ”Long Short-Term Memory”, Neural Computation, 9(8):1735-1780, 1997.

[8] Learning representations by back-propagating errors DE Rumelhart, GE Hinton, RJ Williams Nature 323, 533-536, 1986.

[9] L. Breiman, Random forests, Machine Learning, 45:5-32, 2001.

[10] John Duchi, Elad Hazan, Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, Journal of Machine Learning Research 12, page 2121-2159, 2011.

[11] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, the 3rd International Conference for Learning Representations, San Diego, 2015

[12] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, Samuel Rota Bulo, Deep Neural Decision Forests, The IEEE International Conference on Computer Vision (ICCV), pp. 1467-1475, 2015.

[13] Theano Development Team, ”Theano: A Python framework for fast computation of mathematical expressions”, arXiv e-prints, abs/1605.02688, 2016.

[14] EF Tjong Kim Sang, S Buchholz. “Introduction to the CoNLL-2000 shared task: Chunking”, CoNLL, 2000.

[15] Eric Brill, “A simple rule-based part of speech tagger”, Proceedings of the workshop on Speech and Natural Language Processing (HLT 91), 1991.

[16] Antti-Veikko I Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, Bonnie Dorr, “Combining outputs from multiple machine translation systems”, Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, page 228-235, 2007.

[17] J. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)”, In Proceeding of the IEEE Workshop on Automatic Speech Recognition and Understanding

(ASRU97), Santa Barbara, 1997, pp. 347
[18] Yanjun Ma, Tsuyoshi Okita, Ozlem Cetinoglu, Jinhua Du, Andy Way. Low-Resource Machine Translation Using MaTrEx: The DCU Machine Translation System for IWSLT 2009. International Workshop on Spoken Language Translation (IWSLT 2009), Tokyo, pp.29-36. 2009.