研究レポート　No.15　～アンサンブル学習～

2020 年 12 月 16 日　株式会社アイズファクトリー　https://bodais.com/company/

概要

予測精度を上げることは機械学習において重要な課題である。どれほど高度な手法を用いても、精度の低いモデルは役に立たない。しかし、精度の低いモデルも、複数組み合わせることで高い精度を実現することができる。この手法をアンサンブル学習という。アンサンブル学習は、精度を向上するために欠かせない手法となっており、Kaggle などの機械学習のコンペティションで上位に入賞するためにも必須の技術である。本稿ではアンサンブル学習の代表的な手法として、バギング、ブースティング、スタッキングの 3 つの手法を紹介する。

1.　はじめに

アンサンブル学習とは、複数のモデルを組み合わせることで汎化性能の高いモデルを構築する手法である。最も単純な方法としては、各モデルの予測値の多数決や平均値をとる方法が考えられる。この方法は単純アンサンブル学習と呼ばれる[1]。単一のモデルに比べて、アンサンブル学習は予測精度が高くなることが示されている[2]。アンサンブル学習が、その汎化性能の高さを認められるきっかけになったことの一つに Netflix Prize [3]がある。Netflix Prize とは、2006 年から 2011 年に Netflix 社が開催した大型のコンペティションで、映画の評価データに基づく映画のレコメンドアルゴリズムを題材に、既存のレコメンドシステムの精度を 10%上回ることが条件だった。2009 年にこの条件を達成し優勝したチーム「BellKor's Pragmatic Chaos」がアンサンブル学習を使用していたのである[4]。ここで使用されていた手法は「ブレンディング (blending) [5]」と呼ばれる手法で、Netflix Prize での優勝を機に、基本的な手法として使われるようになった。

本稿では、アンサンブル学習の中でもよく知られる「バギング」、「ブースティング」、「スタッキング」の 3 つの手法に焦点を置いて解説する。先ほどのブレンディングは、スタッキングと非常に似た手法である。

2.　バギング

バギング (bagging) は、単純アンサンブル学習と非常によく似た手法で、複数のモデルを構築し、各モデルの予測を集約して最終的な結果を出力する。結果の出力には多数決や平均値が用いられる。

バギングを特徴づけるのはブートストラップサンプリング (bootstrap sampling) という学習データの抽出手法で、学習に全てのデータを用いるのではなく、モデルごとに復元抽出を行うことで学習データを生成する。この抽出法により、モデルごとの学習データにずれが生じ、データ生成を複数回行った状況を再現するのである。ブートストラップサンプリングとモデルの集約を組み合わせる方法を bootstrap aggregating といい、バギングいう名前も Bootstrap AGGregatING に由来している[6]。

モデルを集約することで、モデルの精度が向上する理由を説明する[1]。l個のモデルを用いて分類学習を行う状況を考える。各モデルの誤り率は一律にθとし、各モデルの出力結果を多数決により集約することで最終的な結果を出力する。このとき、l個のモデルのうちk個のモデルが誤る確率$P(k)$は${}_lC_k\theta^k(1-\theta)^{l-k}$により表すことができ、最終的な結果が誤る確率は$\sum_{k>\frac{l}{2}}^{l}P(k)$により求められる。図 1 に誤り率$\theta=0.3$のモデルを$l=21$個用いた場合の$P(k)$を示す。図

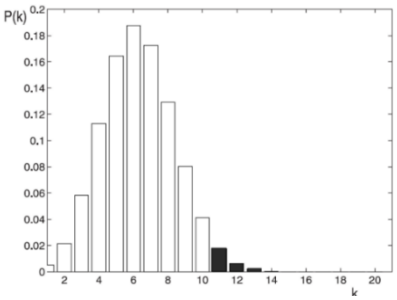

図 1 アンサンブル学習の誤り率[1]

中の黒で塗りつぶされた部分の和が最終的なモデルの誤り率 (0.026) となり、単一のモデルの誤り率 0.3 よりも小さくなることがわかる。

バギングには不安定なモデルを使用すべきことが知られている。不安定というのは、データの変動に対して敏感であることをいい、例えば、SVN のような線形に分類するモデルは安定性が高く、NN のように非線形に分類するモデルは不安定である[7]。不安定なモデルをバギングによって組み合わせることで、バリアンスの小さいモデルを構成することができる。ランダムフォレスト[8]はバギングの代表例である。

3.　ブースティング

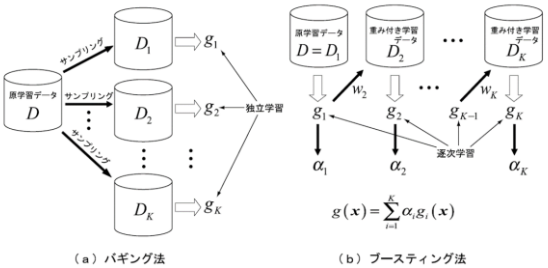

図 2 バギングとブースティング[1]

ブースティング (boosting) とは、複数のモデルを直列に組み合わせることでモデルの精度を向上する手法である。前のモデルが誤って学習したデータに注目して逐次的に学習するため、並列構造のバギングよりも学習に時間がかかる。バギングとブースティングの概要を図 2 に示す。Dが学習データ、gがモデル、αとwはそれぞれ各モデルとデータの重みを表している。

ブースティングの原点は Valiant により提唱された PAC 学習にある[9]。PAC (Probably Approximately Correct) 学習というのは、計算複雑性の概念を学習アルゴリズムに対して導入する研究分野のことで[10]、ブースティングは「PAC 学習における弱学習器を強学習器に変換するにはどうすればいいか」という問題に基づいている[11]。弱学習器とは、ランダムよりは多少精度のよいモデルのこと

で、強学習器というのは、誤分類率が十分小さいモデルのことをいう。弱学習器の精度を boost (向上)することから boosting とよばれる。[11]や[12]などで初期のブースティングの手法が提案されているが、現実的に使用可能な手法としては Adaboost が初めである[9]。

Adaboost のアルゴリズムを簡単に説明する。図 2 にあるように、ブースティングアルゴリズムgはK個の弱学習器g_iを用いて、$g(x)=\Sigma_{i=k}^{K}\alpha_i g_i(x)$により定義されるが、以下の手順でこの$\alpha_i$は決定される。

step 0　N個のデータ$\boldsymbol{x}=\{(x_1,y_1),\ldots.(x_N,y_N)\}$に対して、その重み（生起確率）$D_1=\{w_{1,1},\ldots.w_{1,N}\}$を$w_i=\frac{1}{N}$により初期化する。

step 1　D_iを元にN回の復元抽出を行ったデータで弱学習器g_iを構成する。構成方法はg_iのアルゴリズムによる。

step 2　g_iの誤り率$e_i=\Sigma_{j=1}^{N}w_{i,j}(y_j-g_i(x_j))$を計算する。

step 3　学習器g_iの重み$\alpha_i=\frac{1}{2}\log\frac{1-e_i}{e_i}$を計算する。

step 4　データの重みD_{i+1}を$w_{i+1,j}=\frac{w_{i,j}\exp(-\alpha_i y_j g_i(x_j))}{\Sigma_{k=1}^{N}w_{i,k}\exp(-\alpha_i y_k g_i(x_k))}$により更新する。

step 5　1～4 を$i=K$まで繰り返す。

ブースティングはバイアスが小さくなりやすい一方、バリアンスが大きくなるため過学習が起きやすくなる。一般に、バイアスとバリアンスはトレードオフの関係にあり、「バイアスとバリアンスのジレンマ[1]」として知られている。

4.　スタッキング

スタッキングは、複数のモデルを段階的に学習し、前段のモデルの予測値を後段のモデルの学習に使用する手法である。2 層のスタッキングの概要を図 3 左に示す。θはトレーニングデータ、qは問題データ、Gはモデルとする。1 層目のモデルG_iが学習データθ_0を元に学習し、問題データq_0の予測を出力する。θで学習をしたモデルが問題qに対して出力した予測を$G(\theta;q)$により表すとすると、G_iが出力した予測は$G_i(\theta_0;q_0)$と表せる。1 層目のモデルの予測値を使用して 2 層目のモデルGの学習データθ_1と問題データq_1を構成し、2 層目のモデルから最終予測$p=G(\theta_1;q_1)$を得る。学習データと問題データの構成方法は後述する。最終的なモデルGをメタモデルといい、前段のモデルの信頼性を学習することで最終的な予測を出力している。この前層の出力を次層のモデルが利用するという構成を繰り返すと多層に積み上げる (stack) ことができる。

モデルの予測値を元に、学習データと問題データを構成する方法の概要を説明する[13]。元の学習データをθ、問題データをqで表すとする。

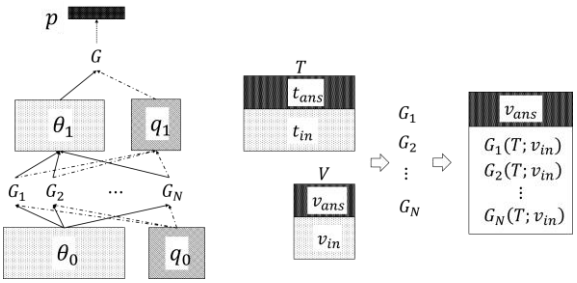

図 3 スタッキングの概要(左)と学習データの構成(右)

step 0　交差検証法の要領で学習データを r 個に分割し、学習データと検証データの組(T_k,V_k)を r 個作る。

step 1　以下の手順で(T_k,V_k)から学習データを構成する。

1. モデルG_iについて学習データT_kによる学習を行い、検証データV_kの入力値$v_{k,in}$を問題データとして予測値$G_i(T_k,v_{k,in})$を出力する。
2. step1-1 をモデル数の N 回繰り返すと、各モデルの予測値$G_i(T_k,v_{k,in})$と正解データ$v_{k,ans}$から次層の学習データが構成できる。$(G_1(T_k,v_{k,in}),\ldots,G_N(T_k,v_{k,in}),v_{k,ans})$

step 2　step1 を r 回繰り返し、各組(T_k,V_k)から学習データを構成する。

step 3　step 1 の要領で各モデルを全データで学習し、問題に対する予測から、次層の問題データ$(G_1(\theta,q),\ldots,G_N(\theta,q))$を構成する。

5.　まとめ

本稿ではアンサンブル学習の代表例として、バギング、ブースティング、スタッキングについて解説した。1 章で紹介したブレンディングのように、この 3 手法以外にも、アンサンブル学習は存在する。

アンサンブル学習によって、モデルの精度向上が期待できるが、データや組み込むモデルによっては必ずしも精度が向上するとは限らない。精度向上の銀の弾丸ではないことは理解する必要がある。

6.　参考文献

[1] 上田修功「アンサンブル学習」コンピュータビジョンとイメージメディア, Vol. 46, No. SIG 15(CVIM 12), p.11-20 (2005)

[2] Thomas G. Dietterich "Machine Learning Research: Four Current Directions" AI Magazine, Vol.18, No. 4, p.97-136 (1997)

[3] Netflix Pirze, https://www.netflixprize.com/

[4] 馬場雪乃「機械学習コンペティションの進展と今後の展開」人工知能 31 巻 2 号 p.248-253, (2016)

[5] Andreas Töscher and Michael Jahrer "The BigChaos Solution to the Netflix Grand Prize" (2009)

[6] Leo Breiman "Bagging Predictors" Machine Learning, Vol. 24, No.2, p.123-140 (1994)

[7] Zhi-Hua Zhou, 訳: 宮岡悦良・下川朝有「アンサンブル法による機械学習 基礎とアルゴリズム」近代科学社 (2012)

[8] Leo Breiman "Random Forests" Machine Learning, Vol. 45, No. 1, p.5-32 (2001)

[9] Yoav Freund, Robert Schapire, 訳: 安倍直樹「ブースティング入門」人口知能学会誌, 14 巻 5 号, (1999)

[10] 篠原歩, 宮野悟「PAC 学習 確率的で近似的に正しい学習」情報処理 Vol. 32, No.3, p. 257-263 (1991)

[11] Yoav Freund "Boosting a weak learning algorithm by majority" (1995)

[12] Pobert E, Schapire "The Strength of Weak Learnability" Machine Learning, 5, p.197-227

[13] David H. Wolpert "Stacked Generalization" Neural Networks, 5(2), p.241-259