PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU
Nr 47 2009
TAKSONOMIA 16
Klasyfikacja i analiza danych – teoria i zastosowania

Eugeniusz Gatnar

Akademia Ekonomiczna w Katowicach

IMPLEMENTACJA METOD ŁĄCZENIA MODELI DYSKRYMINACYJNYCH W PROGRAMIE R

1. Wstęp

Łączenie (agregacja) modeli dyskryminacyjnych stało się już dobrze znanym sposobem poprawy jakości klasyfikacji w praktycznych zastosowaniach. To podejście było rozważane już w XVIII wieku przez francuskiego filozofa Condorceta, który zaproponował twierdzenie o podejmowaniu grupowych decyzji (*Condorcet jury theorem*).

Podejmowanie decyzji większością głosów jest wykorzystywane również w analizie dyskryminacyjnej. Jego zalety widać najlepiej wtedy, gdy łączone są modele jak najbardziej różniące się między sobą. Już w pierwszych praktycznych zastosowaniach zaobserwowano, że poprawa dokładności predykcji modelu zagregowanego zależy od stopnia korelacji modeli indywidualnych.

W artykule zostanie przedstawiony przegląd dostępnych pakietów zawierających dodatkowe procedury napisane w języku **R**. Pokazane zostaną także ich zastosowania w przykładowych programach napisanych w tym języku oraz wyniki analiz porównawczych.

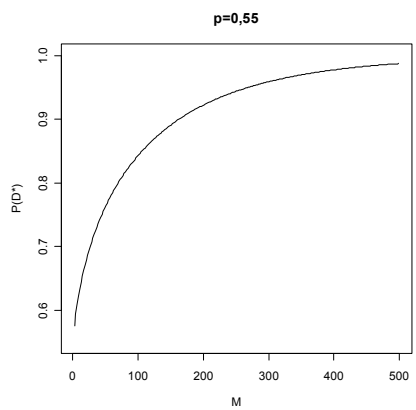
2. Twierdzenie Condorceta

Condorcet [1785] sformułował twierdzenie pozwalające określić prawdopodobieństwo podjęcia trafnej decyzji przez grupę osób. Zakłada ono, że każda z M osób podejmuje trafną decyzję: „tak" lub „nie" z prawdopodobieństwem równym p_i $(i = 1,...,M)$. Można powiedzieć, że to prawdopodobieństwo określa poziom kompetencji *i*-tej osoby do podejmowania decyzji. Jeżeli wszystkie osoby mają ten sam poziom kompetencji (indywidualnej trafności decyzji), tj. $p_1 = ... = p_M = p$, lecz decyzje podejmują niezależnie i większością głosów, to prawdopodobieństwo podjęcia przez nie trafnej decyzji (D^*) można wyznaczyć z rozkładu dwumianowego:

$$p(D^*) = \sum_{i>\frac{M}{2}}^{M} \binom{M}{i} p^i (1-p)^{M-i} . \tag{1}$$

Ponadto jeżeli $p > 0,5$, to $\lim_{M\to\infty} p(D^*) = 1$, co oznacza, że im większa liczba głosujących, tym większe prawdopodobieństwo podjęcia trafnej decyzji.

W miarę wzrostu liczby głosujących następuje szybki wzrost dokładności decyzji podejmowanych kolektywnie, zgodnie z wzorem (1). Na rysunku 1 znajduje się ilustracja faktu, że łączne głosowanie nawet bardzo mało kompetentnych osób ($p = 0,55$) daje wynik tylko nieco lepszy niż rzut monetą, pozwala uzyskać pewność podjęcia trafnej decyzji. Wartości $p(D^*)$ zbliżone do jedności można obserwować po przekroczeniu liczby głosujących $M = 500$.



Rys. 1. Rozkład prawdopodobieństwa $p(D^*)$ dla $p = 0,55$

Źródło: opracowanie własne.

Shapley i Grofman [1984] udowodnili, że twierdzenie Condorceta jest prawdziwe także wtedy, gdy prawdopodobieństwa podjęcia trafnych decyzji przez kolejne osoby (p_i) różnią się od siebie.

Z kolei Krogh i Vedelsby [1995] przedstawili formalny dowód na to, że w przypadku łączenia modeli regresyjnych błąd predykcji modelu zagregowanego D^* zawsze jest mniejszy od średniego błędu modeli składowych.

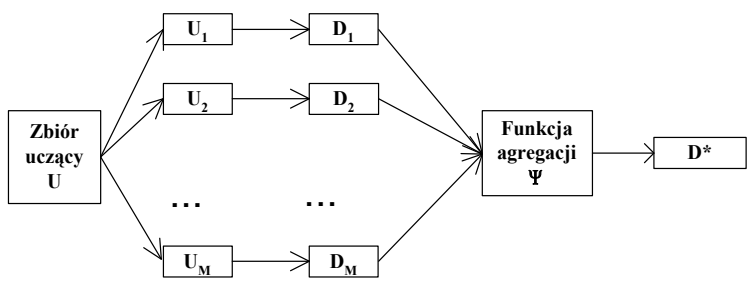
3. Metody łączenia modeli

Łączenie modeli bazowych $D_1,...,D_M$ można w najbardziej ogólny sposób przedstawić jako:

$$\hat{D}^*(\mathbf{x}_i) = \Psi\left(\hat{D}_1(\mathbf{x}_i),...,\hat{D}_M(\mathbf{x}_i)\right), \tag{2}$$

gdzie postać funkcji Ψ zależy od rodzaju wyników predykcji modeli składowych. Aby wynik predykcji modelu zagregowanego (2) był bardziej dokładny niż wynik każdego z modeli składowych, należy wzmacniać wyniki predykcji tych modeli, które dały prawidłowy wynik, osłabiać zaś te, które są błędne.

Na podstawie zbioru uczącego U tworzony jest zbiór M prób uczących $U_1,...,U_M$. Próby te mogą zawierać albo podzbiory obserwacji wybranych ze zbioru uczącego U, albo wszystkie obserwacje, lecz charakteryzowane przez niektóre zmienne. Próby $U_1,...,U_M$ służą do budowy wspomnianych modeli bazowych $D_1,...,D_M$, których wyniki podlegają agregacji. W większości znanych propozycji proces łączenia modeli w podejściu wielomodelowym odbywa się tak, jak to przedstawiono na rys. 2.



Rys. 2. Model zagregowany D^*

Źródło: opracowanie własne.

Najbardziej popularną metodą łączenia modeli dyskryminacyjnych jest głosowanie większością głosów. Innymi słowy, oznacza to, że model zagregowany D^* przydziela obserwację $\mathbf{x}_i$ do tej klasy C_j, którą wskazała większość spośród modeli bazowych $D_1,...,D_M$:

$$\hat{D}^*(\mathbf{x}_i) = \arg\max_j \sum_{m=1}^{M} I(\hat{D}_m(\mathbf{x}_i) = C_j). \tag{3}$$

Tumer i Ghosh [1996] udowodnili, że błąd klasyfikacji modelu zagregowanego D^* maleje w miarę spadku stopnia podobieństwa modeli składowych $D_1,...,D_M$. A więc należy łączyć tylko takie modele, których wyniki klasyfikacji tych samych obserwacji jak najbardziej różnią się od siebie.

Aby zapewnić wysoki poziom zróżnicowania łączonych modeli składowych, stosuje się różne rozwiązania. Do najbardziej znanych należą:

- dobór obserwacji ze zbioru uczącego U do prób uczących $U_1,...,U_M$,
- dobór zmiennych objaśniających $X_1,...,X_L$ do poszczególnych modeli bazowych,

- przekształcanie (przekodowanie) wartości zmiennej zależnej,
- zmiana parametrów poszczególnych modeli bazowych (np. liczby węzłów drzewa klasyfikacyjnego),
- budowa modeli bazowych za pomocą odmiennych metod, np. sieci neuronowe, metoda wektorów nośnych czy drzewa klasyfikacyjne.

Jeśli chodzi o to pierwsze rozwiązanie, to polega ono na przygotowaniu prób uczących $U_1,...,U_M$, np. za pomocą losowania obserwacji ze zbioru U. Daje to bardzo dobre rezultaty w przypadku modeli niestabilnych, takich jak chociażby drzewa klasyfikacyjne. Najbardziej popularną metodą tego typu jest metoda losowania boostrapowego (*bagging*) zaproponowana przez Breimana [1996].

Według innej propozycji, znanej pod nazwą *boosting* (wzmacnianie), a przedstawionej przez Freunda i Schapire'a [1996], obserwacje są losowane do prób uczących na podstawie wag proporcjonalnych do błędu klasyfikacji charakteryzującego odpowiedni model składowy D_m.

4. Pakiety w programie R

Metody łączenia modeli dyskryminacyjnych i regresyjnych stały się już tak popularne wśród statystyków, że znalazło to wyraz w dostępnych programach realizujących obliczenia statystyczne.

Jednak w komercyjnych produktach metody agregacji modeli są bardzo rzadko implementowane (np. SAS, SPSS). Jedynie w programie *STATISTICA* występuje moduł DRZEWA KLASYFIKACYJNE I REGRESYJNE ZE WZMACNIANIEM (*boosted trees*). Natomiast w systemie *STATISTICA Data Miner* jest moduł RANDOM FORESTS.

Natomiast w programie **R** dostępnych jest 6 pakietów służących do łączenia modeli dyskryminacyjnych: `ada`, `adabag`, `boost`, `gbm`, `ipred` oraz `randomForest`. Zostaną one szerszej omówione dalej.

4.1. Pakiet `ada`

Autorami tego pakietu są M. Culp, K. Johnson i G. Michailidis z Uniwersytetu w Michigan, a jego najnowsza wersja powstała w dniu 16 lutego 2008 r. Znajdują się w nim procedury realizujące stochastyczną metodę *AdaBoost* w dyskryminacji (dla dwóch klas) i regresji, zaproponowaną przez Friedmana i in. [2000]. Modele bazowe mają postać drzew klasyfikacyjnych i regresyjnych budowanych za pomocą procedury `rpart`.

Składnia podstawowego polecenia służącego do budowy modelu została przedstawiona poniżej:

```
ada(x, y, test.x, test.y=NULL, loss=c("exponential",
"logistic"),
```

```
type=c("discrete", "real"), iter=50, nu=0.1,
bag.frac=model.coef=TRUE, bag.shift=FALSE, max.iter=20,
delta=10^(-10), …).
```

4.2. Pakiet `adabag`

Pakiet ten został zbudowany przez E. Cortes, M. Martineza i N. Rubio (Hiszpania), a jego najnowsza wersja pochodzi z dnia 16 lutego 2008 r. Umożliwia on agregację modeli dyskryminacyjnych za pomocą metody *AdaBoost* zaproponowanej przez Freunda i Schapire'a [1996] oraz metody *bagging*, którą przygotował Breiman [1996]. Modelami bazowymi w tym pakiecie są drzewa klasyfikacyjne tworzone przez procedurę `rpart`.

Polecenia odpowiadające obu metodom mają następującą składnię:

```
adaboost.M1(formula, data, boos=TRUE, mfinal=100, coef-
learn='Breiman',
minsplit=5, cp=0.01)
bagging(formula, data, mfinal=100, minsplit=5,
cp=0.01).
```

4.3. Pakiet `boost`

Autorem tego pakietu, którego najnowsza wersja pochodzi z 16 lutego 2008 r., jest M. Dettling z John Hopkins University. Pakiet zawiera funkcje *BagBoost*, *LogitBoost*, *AdaBoost* i *L2Boost* realizujące metody agregacji modeli dyskryminacyjnych (dla dwóch klas) i regresyjnych przedstawione w pracy Dettlinga i Bühlmanna [2003].

Polecenie realizujące metodę wzmacniania (*boosting*) ma następującą składnię:

```
adaboost(xlearn, ylearn, xtest, presel=200, mfinal=100).
```

4.4. Pakiet `gbm`

Pakiet ten, przygotowany przez Ridgewaya [1999] z RAND (jego najnowsza wersja datowana jest na 3 sierpnia2007 r.), zawiera implementacje algorytmu *AdaBoost* oraz stochastycznej metody *boosting* Friedmana [2002] do agregacji modeli dyskryminacyjnych i regresyjnych. W tym ostatnim przypadku można wykorzystywać różne funkcje straty: logistyczną, Poissona, Coksa oraz wykładniczą.

Składnia polecenia, które realizuje budowę wszystkich rodzajów modeli zagregowanych, ma postać:

```
gbm(formula=formula(data), distribution="bernoulli",
data=list(),
weights, var.monotone=NULL, n.trees=100, interac-
tion.depth=1,
n.minobsinnode=10, shrinkage=0.001, bag.fraction=0.5,
```

```
train.fraction=1.0, cv.folds=0, keep.data=TRUE, ver-
bose=TRUE).
```

4.5. Pakiet `ipred`

Ten pakiet przygotowali A. Peters i T. Hothorn z Uniwersytetu w Erlangen, a jego najnowsza wersja pochodzi z 30 lipca 2008 r. Znajduje się w nim implementacja metody *bagging* Breimana [1996], którą można zastosować w dyskryminacji i regresji, oraz metody *bundling*, zaproponowanej przez Hothorna i Lausena [2003], która wykorzystuje dodatkowe zmienne objaśniające. Są one wynikiem predykcji modeli zbudowanych na podstawie zbioru obserwacji, które nie znalazły się w próbie bootstrapowej U_m (*out-of-bag*). Predykcja jest dokonywana dla obserwacji z próby uczącej U_m, a modele bazowe w tej metodzie mają postać drzew klasyfikacyjnych, dlatego wymagane jest zainstalowanie także pakietu `rpart`.

Składnia polecenia, która realizuje agregację modeli metodą *bagging*, ma postać:

```
bagging(formula, data, subset, na.action=na.rpart, ...).
```

Natomiast agregacja modeli metodą *bundling* polega na użyciu dwóch poleceń:

```
comb.lda<-list(list(model=lda, predict=function(obj, nd)
predict(obj, nd)$x))
bagging(formula, data, comb=comb.lda).
```

4.6. Pakiet `randomForest`

W tym pakiecie znajduje się implementacja oryginalnego algorytmu Breimana [2001] do agregacji drzew klasyfikacyjnych wykorzystującej losowy dobór zmiennych do budowy modeli składowych. Oryginalny program w języku FORTRAN napisali L. Breiman i A. Cutler, natomiast połączenie z programem **R** zrealizowali A. Liaw i M. Wiener. Najnowsza wersja tego pakietu pochodzi z 17 kwietnia 2008 r., a modelami bazowymi są drzewa klasyfikacyjne.

Składnia podstawowego polecenia realizującego agregację modeli ma postać :

```
randomForest(x, y, xtest, ytest, ntree=500, replace=TRUE,
classwt=NULL, cutoff, strata, nodesize=5, importance=FALSE,
localImp=FALSE, nPerm=1, proximity, oob.prox=proximity,
norm.votes=TRUE, do.trace=FALSE, corr.bias=FALSE,
keep.inbag=FALSE, ...).
```

Większość z pakietów omówionych powyżej wykorzystuje drzewa klasyfikacyjne i regresyjne jako modele bazowe. W tym przypadku następuje automatyczne wywołanie procedury `rpart` przygotowanej przez Therneau i Atkinsona [1997]. Możliwe jest więc wykorzystanie dodatkowych parametrów sterujących budową drzew, np. za pomocą polecenia `rpart.control`.

5. Analiza porównawcza

W ramach eksperymentu obliczeniowego przygotowano porównania wielkości czasu pracy komputera w zadaniu agregacji modeli dyskryminacyjnych jedynie za pomocą metody *bagging* dla różnych zbiorów danych.

Prosta procedura napisana w języku **R** dla tej metody, która daje wynik predykcji dla obserwacji ze zbioru `zbior.rozpoznawany`, ma postać:

```
tablica <- NULL
m <- nrow(zbior.uczacy)
for (i in 1:lmod)
{
ucz <- sample(1:m, size = m, replace = T)
proba.uczaca <- zbior.uczacy[ucz, ]
model.bazowy <- rpart(klasa~., proba.uczaca)
# drzewo klasyfikacyjne
wynik <- predict(model.bazowy, zbior.rozpoznawany, type
="class")
tablica <- cbind(tablica, as.matrix(wynik))
}
for (i in 1:lobs) predykcja[i] <- glosuj(tablica[i, ]).
```

Porównanie szybkości pracy procesora (w sekundach) w przypadku agregacji 50 modeli bazowych, które miały postać drzew klasyfikacyjnych, dla wybranych zbiorów danych znajduje się w tab. 1. Zastosowano odpowiednie procedury z przedstawionych powyżej pakietów oraz procedurę własną, a czas ich pracy był mierzony za pomocą polecenia `system.time(bagging)`.

Tabela 1. Czas pracy procesora dla metody *bagging*

Zbiór danych	Własna procedura	Pakiet `adabag`	Pakiet `ipred`
DNA	15,4	5,4	4,8
Letter	21,3	9,8	9,2
Satellite	12,0	5,6	5,1
Spam	11,3	8,8	7,9
Zip	13,9	6,4	6,9

Źródło: opracowanie własne na podstawie baz danych z Uniwersytetu Kalifornijskiego.

Podobne porównania czasu pracy procesora wykonano również dla metody *boosting*, chociaż nie wszystkie pakiety realizują klasyczną jej wersję, zaproponowaną w pracy Freunda i Schapire [1996].

6. Konkluzje

Program **R** jest najbardziej dogodnym środowiskiem do budowy modeli zagregowanych: można samodzielnie pisać odpowiednie procedury albo korzystać z

kilku dostępnych pakietów. Zawsze jednak użytkownik w pełni kontroluje przebieg procesu łączenia modeli.

Najlepsze ze wspomnianych pakietów realizujących agregację modeli różnymi metodami to: `adabag`, `ipred` i `randomForest`, ponieważ wykorzystują one kod napisany w językach C i FORTRAN. To wyraźnie zwiększa szybkość pracy, a zastosowane algorytmy są bardzo stabilne.

Literatura

Breiman L. (1996), *Bagging predictors*, „Machine Learning" no 24, s. 123-140.

Breiman L. (2001), *Random forests*, „Machine Learning" no 45, s. 5-32.

Condorcet, Marquis de (1785), *Essais sur l'application de l'analyse a la probabilite des decisions redues a la pluralite des voix*, Paris.

Dettling M., Bühlmann P. (2003), *Boosting for tumor classification with gene expression data*, „Bioinformatics", tom 19, s. 1061-1069.

Freund Y., Schapire R. (1996), *A decision-theoretic generalization of on-line learning and application to boosting*, „Journal of Computer and System Sciences" no 55, s. 119-139.

Friedman J. (2002), *Stochastic gradient boosting*, „Computational Statistics and Data Analysis", tom 38(4), s. 367-378.

Friedman J., Hastie T., Tibshirani R. (2000), *Additive logistic regression: a statistical view of boosting*, „Annals of Statistics" no 28(2), s. 337-407.

Hansen L.K., Salamon P. (1990), *Neural network ensembles,* IEEE Transactions on Pattern Analysis and Machine Intelligence, t. 12, s. 993-1001.

Hothorn T., Lausen B. (2003), *Bundling classifiers by bagging trees,* „Computational Statistics & Data Analysis" 2005, tom 49, s. 1068-1078.

Krogh A., Vedelsby J. (1995), *Neural network ensembles, cross validation, and active learning*, [w:] G. Tesauro, D. Touretzky, T. Leen (eds.), *Advances in Neural Information Processing Systems*, MIT Press, 7, s. 231-238.

Ridgeway G. (1999), *The state of boosting*, „Computing Science and Statistics" tom 31, s. 172-181.

Shapley L., Grofman B. (1984), *Optimizing group judgemental accuracy in the presence of interdependences*, „Public Choice" no 43, s. 329-343.

Therneau T.M., Atkinson E.J. (1997), *An introduction to recursive partitioning using the RPART routines*, Mayo Foundation, Rochester.

Tumer K., Ghosh J. (1996), *Analysis of decision boundaries in linearly combined neural classifiers*, „Pattern Recognition" no 29, s. 341-348.

THE IMPLEMENTATION OF ENSEMBLE METHODS IN R

Summary

Model aggregation is a well known technique used to improve the classification accuracy in many applications.

In this paper, we review a number of available packages in the **R** environment that can be used for aggregation of classification models. We also compare the CPU time when procedures from different packages were applied. The comparison was done for five data sets from the UCI Repository.