

Wykład 6 Wzmacnianie klasyfikatorów

dr hab. Konrad Furmańczyk, prof. SGGW

Instytut Informatyki Technicznej/KZM, SGGW, bud. 34, pok 3/87
email: konrad_furmanczyk@sggw.edu.pl

November 9, 2021

Na podstawie

<https://www.listendata.com/2014/11/random-forest-with-r.html>

(randomForest)

```
rf <- randomForest(Creditability~., data=mydata, ntree=500)
```

```
print(rf)
```

Type of random forest: classification Number of trees: 500

No. of variables tried at each split: 4

OOB estimate of error rate: 23.1%

Confusion matrix: 0 1 class.error

0 131 169 0.56

1 62 638 0.09

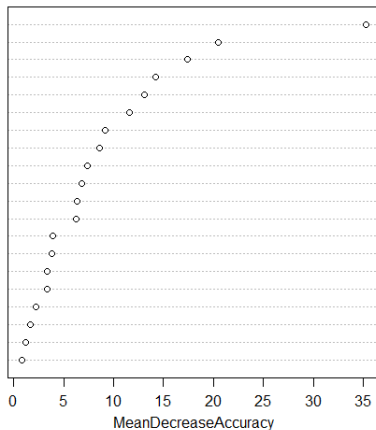
```
rf <-randomForest(Creditability~.,data=mydata, mtry=best.m,  
importance=TRUE,ntree=500)  
print(rf)
```

```
importance(rf)
```

```
varImpPlot(rf)
```

<https://blakelobato1.medium.com/k-nearest-neighbor-classifier-implement-homemade-class-compare-with-sklearn-import-6896f49b89e>

Account.Balance
Duration.of.Credit.month.
Payment.Status.of.Previous.Credit
Credit.Amount
Value.Savings.Stocks
Guarantors
Most.valuable.available.asset
Age..years.
Length.of.current.employment
Purpose
No.of.Credits.at.this.Bank
Instalment.per.cent
Type.of.apartment
Concurrent.Credits
No.of.dependents
Occupation
Duration.in.Current.address
Telephone
Foreign.Worker
Sex...Marital.Status



Mean Decrease Accuracy - mierzy różnice "accuracy" jeśli usuniemy tę zmienną.

Mean Decrease Gini - na podstawie indeksu Giniego dla podziałów drzew.

```
pred1=predict(rf,type = "prob")  
library(ROCR)  
perf = prediction(pred1[,2], mydata$Creditability)  
AUC  
auc = performance(perf, "auc")  
auc
```

ROC

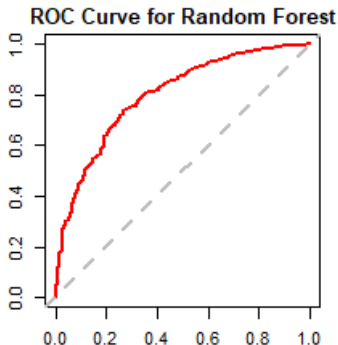
True Positive and Negative Rate

```
pred3 = performance(perf, "tpr","fpr")
```

Plot the ROC

```
curve plot(pred3,main="ROC Curve for Random  
Forest",col=2,lwd=2)
```

```
abline(a=0,b=1,lwd=2,lty=2,col="gray")
```



Wzmacnianie klasyfikatorów za pomocą bootstrapu

n elementowa próba bootstrapowa to próba powstała w wyniku n -krotnego losowania pojedynczej obserwacji z próby uczącej ze zwracaniem.

W losowaniu ze zwracaniem średnio $1/3$ elementów nie zostaje wylosowana do psuudopróby o liczebności n , bowiem prawdopodobieństwo niewylosowania danej obserwacji wynosi $\left(1 - \frac{1}{n}\right)^n \approx \frac{1}{e} = 0.368$.

1. Dla $b = 1, \dots, B$ losujemy próbę bootstrapową \mathcal{L}^* z próby uczącej \mathcal{L} i konstruujemy klasyfikator d_b dla próby bootstrapowej \mathcal{L}^*
2. Obserwacje o wektorze cech \mathbf{x} klasyfikujemy wg. reguły głosowania większościowego

$$d_{bag1}(\mathbf{x}) = \underset{1 \leq k \leq K}{\operatorname{argmax}} \sum_{b=1}^B I(d_b(\mathbf{x}) = k).$$

37% elementów próby uczącej nie zostanie włączona do próby bootstrapowej. Na tych elementach można wyznaczyć błędy klasyfikatorów bootstrapowych d_b i przypisać wagi do tych klasyfikatorów. Większe wagi dla tych klasyfikatorów klasyfikatorów o większym odsetku poprawnych klasyfikacji.

Obliczamy $w^{(b)} = \sum_{j=1}^n I(\mathbf{x}_j \notin \mathcal{L}^*) I(d_b(\mathbf{x}_j) = y_j) / \sum_{j=1}^n I(\mathbf{x}_j \notin \mathcal{L}^*)$

Normalizacja $w^{(b)} = w^{(b)} / \sum_{b=1}^B w^{(b)}$

Klasyfikator

$$d_{bag2}(\mathbf{x}) = \underset{1 \leq k \leq K}{\operatorname{argmax}} \sum_{b=1}^B w^{(b)} I(d_b(\mathbf{x}) = k).$$

Używamy dla poprawy niestabilnych metod klasyfikacji (np. drzew klasyfikacyjnych). Może znacząco zredukować poziom wariancji niestabilnych klasyfikatorów.

Nie zostaje zachowana struktura modelu np. dla drzew klasyfikacyjnych. Tracimy prostotę interpretacji modelu (drzewa).

Pewne uogólnienia do boosting (AdaBoost, różna warianty: Gradient Boosting (GBM), Extreme Gradient Boosting (XGboost), LightGBM, CatBoost). Korzysta z prób bootstrapowych.

Algorithm 10.1 *AdaBoost.M1*.

1. Initialize the observation weights $w_i = 1/N$, $i = 1, 2, \dots, N$.
2. For $m = 1$ to M :
 - (a) Fit a classifier $G_m(x)$ to the training data using weights w_i .
 - (b) Compute
$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
 - (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
 - (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$, $i = 1, 2, \dots, N$.

3. Output $G(x) = \text{sign} \left[\sum_{m=1}^M \alpha_m G_m(x) \right]$.
-

Łączenie klasyfikatorów d_1, \dots, d_c metodą głosowania

$$d_N(\mathbf{x}) = \operatorname{argmax}_{1 \leq k \leq K} \sum_{j=1}^c I(d_j(\mathbf{x}) = k).$$

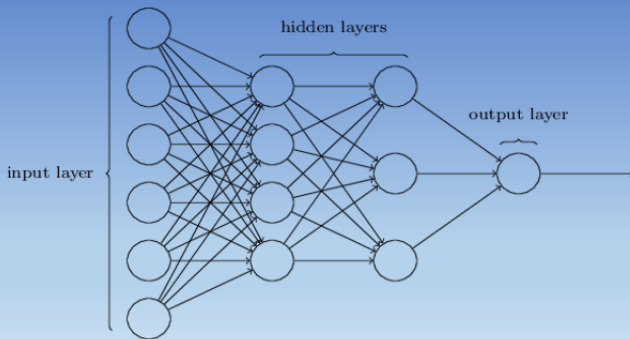
Ważenie klasyfikatorów w zależności od jakości klasyfikatorów

$$d_N(\mathbf{x}) = \operatorname{argmax}_{1 \leq k \leq K} \sum_{j=1}^c w_{jk} I(d_j(\mathbf{x}) = k),$$

gdzie w_{jk} -wagi, np.

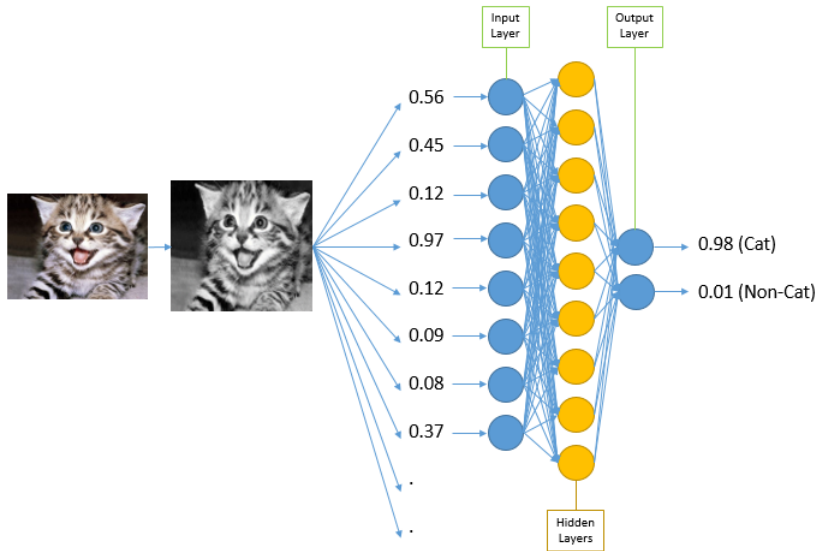
$$w_{jk} = \sum_{i=1}^n I(d_j(\mathbf{x}) = k, y_i = k) / \sum_{i=1}^n I(y_i = k).$$

Classification of Neural Networks

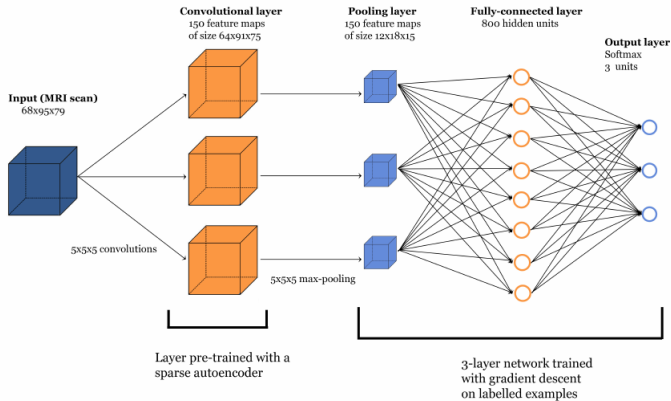


educba.com

Sieci neuronowe



Sieć konwolucyjna



Zbiór danych nazywamy niezbalansowanym (ang. unbalanced) jeżeli zawiera znacznie więcej obserwacji z jednej klasy (ang. majority class) niż z pozostałych lub jedna z klas (ang. minority class) zawiera dużo mniej obserwacji niż pozostałe.

Dokładność (accuracy) klasyfikacji może być bardzo dobra na klasach większościowych i bardzo słaba na klasach mniejszościowych.

Metody:

Undersampling majority class – losowe zmniejszanie liczebności dużych klas.

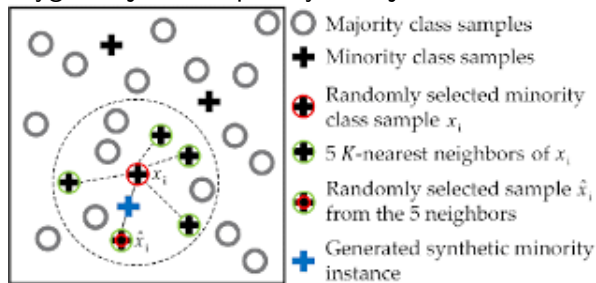
Oversampling minority class – zwiększanie liczebności małych klas (dodajemy losowo kopie klas mniejszościowych). Dodanie sztucznych danych – podobne do oversamplingu, ale nowe dane nie są kopiami starych.

SMOTE (ang. Synthetic Minority Oversampling Technique) i jej wariant ADASYN (ang. Adaptive Synthetic Sampling Method).

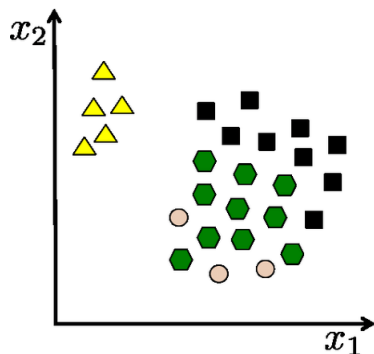
Znajdź dla każdej obserwacji (dla każdego cech z osobna) z klasy mniejszościowej k -najbliższych sąsiadów.

Wyznacz proste z wybranego punktu do wszystkich k sąsiadów.

Wygeneruj losowo punkty na tej linii.



Klasyfikacja wieloetykietowa



$$K = 2$$

$$L = 3$$

$$y_j \in \{0, 1\}$$

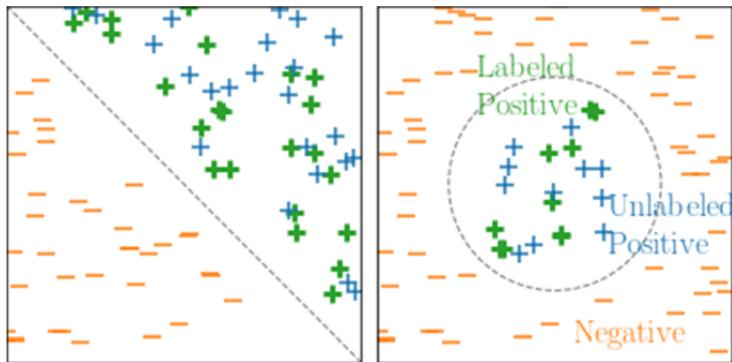
$$\text{light orange circle} \rightarrow \mathbf{y} = [1, 0, 0]$$

$$\text{black square} \rightarrow \mathbf{y} = [0, 1, 0]$$

$$\text{green hexagon} \rightarrow \mathbf{y} = [1, 1, 0]$$

$$\text{yellow triangle} \rightarrow \mathbf{y} = [0, 0, 1]$$

PU learning



Obserwacje mogą należeć do wielu klas

Chory pacjent ma wiele różnych chorób.

Produkt w sklepie klasyfikujemy do wielu klas jednocześnie:
tani/drogi, energooszczędny (tak/nie), modny (tak/nie), np. tani,
energooszczędny i modny lub drogi, energooszczędny i nie modny.

Obserwujemy w pełni etykiety klasy pozytywnej, oraz mamy obserwacje bez etykiet (mieszanina obserwacji z klasy pozytywnej i negatywnej).

Przykłady: fake newsy, chory pacjent zdiagnozowany i pacjent bez diagnozy (może być zdrowy lub chory), preferencje odwiedzania stron internetowych (lubimy strony które odwiedziliśmy, nie wiem czy polubimy strony których nie widzieliśmy).