Evaluarea performantei

Ruxandra Stoean
rstoean@inf.ucv.ro
http://inf.ucv.ro/~rstoean

Bibliografie

- N. Japkowicz, M. Shah, Evaluating Learning Algorithms, Cambridge University Press, 2011
- Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York (2001)
- Catalin Stoean, Ruxandra Stoean, Support Vector Machines and Evolutionary Algorithms for Classification: Single or Together?, Intelligent Systems Reference Library, Volume 69, Springer, 2014.

Directii

- Masuri de performanta
- Estimarea erorii
- Testarea semnificatiei statistice

Acuratete / Rata erorii

- Cele doua masuri complementare dau o masura a performantei generale a clasificatorului.

- Acuratetea se calculeaza ca raportul dintre numarul de date etichetate corect impartit la numarul total de exemple.
  - Rata erorii este raportul dintre numarul de date etichetate gresit impartit la numarul total de exemple.

- Dezavantaje in cazurile:
  - Clase neechilibrate ca numar de date pentru fiecare (class imbalance)
  - Costuri diferite de clasificare gresita pentru fiecare clasa in parte (different misclassification costs)

Masuri statistice de concordanta (agreement)

- Se masoara ponderea coincidentei in concordanta dintre etichetarile clasificatorului si adevaratele iesiri ale datelor.
- Cel mai utilizata statistica: Cohen's k (kappa).
- Dupa Landis JR, Koch GG (1977) Biometrics, 33: 159-174, valoarea k rezultata indica:
  - k < 0 - "No agreement"
  - 0 < k < 0.2 - "Slight agreement"
  - 0.2 < k < 0.4 - "Fair agreement"
  - 0.4 < k < 0.6 - "Moderate agreement"
  - 0.6 <k < 0.8 - "Substantial agreement"
  - 0.8 < k < 1.0 - "Almost perfect agreement"

Matricea de confuzie (Confusion matrix)

- Pentru k clase, este o matrice de k × k
  - Elementul de la pozitia (i, j) da numarul de date care au in BD clasa i si clasificatorul le eticheteaza cu clasa j.
- Tabelul ilustreaza cazul particular pentru doua clase (o etichete pozitive si negative).
- N = TN + FP, P = FN + TP

	Prognozat negative	**Prognozat pozitive**
Real negative	True negative (TN)	False positive (FP)
Real pozitive	False negative (FN)	True pozitive (TP)

Masuri de performanta catre o singura clasa de interes 1/2

- Clasa de interes e numita pozitiva.
- True-positive rate (TPR) refera proportia de date de clasa *i* care sunt etichetate drept clasa i si de catre clasificator.
- False-positive rate (FPR) – proportia de date care nu apartin clasei *i* dar sunt etichetate cu aceasta clasa.
- Formulele pentru doua clase:
- $TPR = \frac{TP}{TP+FN}$
- $FPR = \frac{FP}{FP+TN}$

Masuri de performanta catre o singura clasa de interes 2/2

- In cazul binar, se pot defini cele doua masuri si pentru clasa opusa, negativa.
- True- negative rate $TNR = \frac{TN}{TN+FP}$
- False-negative rate $FNR = \frac{FN}{FN+TP}$

- TPR mai poarta numele de sensitivity (sau recall).
- TNR se mai numeste si specificity.

Rata de verosimilitate (Likelihood ratio)

- Combina sensitivity and specificity pentru a vedea masura in care clasificatorul e eficient in a face o prognoza asupra celor doua clase.
- $LR_+ = \frac{Sensitivity}{1 - Specificity} \rightarrow max$
- $LR_- = \frac{1 - Sensitivity}{Specificity} \rightarrow min$
- LR_+ calculeaza de cate ori este mai mult probabil ca obiecte pozitive sa aiba o predictie pozitiva fata de obiectele negative.
- LR_- calculeaza de cate ori este mai putin probabil ca obiecte pozitive sa aiba o predictie negativa fata de obiectele negative.

LR in compararea a doi algoritmi

- Doi algoritmi A1 si A2.
- Mai intai, $LR_+ \geq 1$
  - In caz contrar, se inverseaza LR_+ cu LR_-.
- Daca $LR_+(A1) > LR_+(A2)$ si $LR_-(A1) < LR_-(A2)$ atunci A1 este superior in intregime.
- Daca $LR_+(A1) < LR_+(A2)$ si $LR_-(A1) < LR_-(A2)$ atunci A1 este superior in confirmarea exemplelor negative.
- Daca $LR_+(A1) > LR_+(A2)$ si $LR_-(A1) > LR_-(A2)$ atunci A1 este superior in confirmarea exemplelor pozitive.

Positive predictive value (PPV)

- Mai poarta numele si de precision.
- Este proportia de date care apartin clasei *i* din toate cele detectate de algoritm ca fiind clasa *i*.
- Pentru cazul binar:
- $PPV = \frac{TP}{TP+FP}$
- si reversul NPV=$\frac{TN}{TN+FN}$
- Valoarea PPV=a spune ca o predictie pozitiva a clasificatorului respectiv va fi adevarata doar in a% din cazuri.
- Masura F (F measure) combina precision si recall intr-o singura masura drept media armonica ponderata (printr-un parametru α) a celor doua.

Analiza ROC

- Receiver operating characteristic (ROC)
- Curba ROC pentru un clasificator este un grafic in care pe axa orizontala avem FPR iar pe axa verticala TPR.
- ROC studiaza asadar relatia dintre sensitivity si specificity pentru un clasificator.
- Spatiul ROC – este un patrat de latura 1.

Spatiul ROC 1/2

- Iesirea unui clasificator determina un punct in spatiul ROC.
  - (0,0) semnifica un clasificator ce eticheteaza toate datele ca negative (TPR = FPR = 0).
  - (1,1) semnifica un clasificator care le eticheteaza pe toate ca pozitive (TPR = FPR = 1).
  - Clasificatorii ale caror iesiri se pozitioneaza pe diagonala principala (TPR = FPR) eticheteaza datele in mod aleatoriu.
  - Cei care se regasesc deasupra diagonalei principale au o performanta mai buna decat aleatoriul.
    - Cei de sub diagnonala au o performanta mai slaba decat aleatoriul.

Spatiul ROC 2/2

- Pentru doua puncte in spatiul ROC p1 si p2, p1 este un clasificator mai bun decat p2 daca p1 este la stanga lui p2 si mai sus decat cel din urma.
- (1,0) (FPR = 1, TPR = 0) semnifica un clasificator care are numai predictii gresite.
- (0,1) semnifica clasificatorul ideal (eticheteaza toate datele pozitive corect si nu greseste in etichetarea exemplelor negative).
- Clasificatoarele cu iesire pe diagonala secundara au o performanta egala pe exemplele pozitive cat si pe cele negative.

Curba ROC

- Un operating point corespunde unui prag de decizie prin care clasificatorul discrimineaza datele ca fiind pozitive sau negative.
- Datele cu un scor peste acest prag sunt etichetate ca fiind pozitive, iar cele cu un scor sub sunt considerate negative.
- Prin varierea pragului intre scorul minim si maxim al datelor se obtine cate un TPR si un FPR care pot fi plotate, rezultand in final curba ROC.

Area under the ROC curve (AUC)

- AUC – masura a performantei unui clasificator.
- Valoarea AUC apartine intervalului [0,1]
  - 1 este valoarea pentru un clasificator perfect.
  - Valoarea AUC pentru un clasificator aleator este 0.5.
  - Pentru o performanta rezonabila, un clasificator trebuie sa aiba un AUC peste 0.5.

Estimarea erorii 1/2

- Se realizeaza pentru o estimare obiectiva a performantei unui clasificator.
- Datele se impart in trei parti:
  - O multime de antrenament din care clasificatorul invata asocierile dintre atribute si clase.
  - O multime de validare pentru a determina eroarea de predictie a modelului (numai cand colectia de date e suficient de mare)
  - O multime de test pentru a masura eroarea de generalizare a abordarii.
  - Pentru ultimele doua multimi, iesirea datelor este considerata a fi necunoscuta.

Estimarea erorii 2/2

- Cross-validation estimeaza acuratetea de predictie pe care modelul o va arata in practica.
- Se selecteaza multimi de antrenament si de test de un numar de ori dupa mai multe scheme posibile.
- Abilitatea de generalizare a clasificatorului se verifica calculand acuratetea de predictie pe multimea de test ca medie a mai multe runde de cross-validation.
- Daca multimea de date permite, se estimeaza eroarea de predictie pe multimea de validare, anterior procedurii pe multimea de test.

Random subsampling

- Cea mai simpla tehnica de resampling.
- Se executa n rulari ale clasificatorului
  - Uzual $n = 30$.
- La fiecare rulare se impartea multimea de date:
  - 2/3 multime de antrenament
  - 1/3 multime de test
  - Repartitia datelor in cele doua multimi se face aleator.
- Acuratetea de predictie finala este media valorilor obtinute in cele 30 de rulari pe multimile de test.

Testarea semnificatiei statistice 1/2

- Utila (si obligatorie) cand:
  - Performanta unui algoritm nou se compara cu cele ale algoritmilor standard.
  - Se testeaza care e cel mai potrivit algoritm pentru o problema data.
- Se executa n ($n = 30$) rulari ale algoritmilor prin cross-validation.
- Se presupune ipoteza nula H_0 – algoritmii testati au o performanta similara.

Testarea semnificatiei statistice 2/2

- Ipoteza nula H_0 se respinge daca valoarea p (p-value) intoarsa de test este mai mica sau egala cu nivelul de semnificatie (significance level) de 0.05.
- Optiunea sigura de testare a semnificatiei statistice este utilizarea unui test nonparametric care nu face presupuneri privind distributia masurilor de performanta.
- Wilcoxon's signed-rank test for matched pairs (in circumstante similare).

Exemplificare in R 1/4

```
library(e1071)
library(rpart)
library(mlbench)
library(fmsb) # pentru Cohen's k
library(ROCR) # pentru ROC
library(stats) # pentru Wilcoxon's test


data(PimaIndiansDiabetes)
dat <- PimaIndiansDiabetes

repeats <- 30
classColumn <- 9
accuraciesSVM <- vector(mode="numeric",length=10)
accuraciesDT <- vector(mode="numeric",length=10)
index <- 1:nrow(dat)
```

Exemplificare in R 2/4

```
for(i in 1:repeats){
  # generare aleatoare a multimilor de antrenament si test
  testindex <- sample(index, trunc(length(index)/4))
  testset <- dat[testindex, ]
  trainset <- dat[-testindex, ]
  # antrenare cu SVM
  svm.model <- svm(diabetes ~ ., data = trainset, kernel = "linear", cost = 1, probability  =TRUE)
  svm.pred <- predict(svm.model, testset[, -classColumn])
  # antrenare si cu DT (pentru comparatie)
  rpart.model <- rpart(diabetes ~ ., data = trainset,  method="class")
  rpart.pred <- predict(rpart.model, testset[, -classColumn], type = c("class"))
  # construirea matricelor de confuzie
  contabSVM <- table(pred = svm.pred, true = testset[, classColumn])
  contabDT <- table(pred = rpart.pred, true = testset[, classColumn])
  #acuratetea obtinuta de SVM, respectiv DT, in fiecare rulare
  accuraciesSVM[i] <- classAgreement(contabSVM)$diag
  accuraciesDT[i] <- classAgreement(contabDT)$diag
} #cross-validation n = 30 de tip random subsampling
```

Exemplificare in R 3/4

```
print(accuraciesSVM)
print(accuraciesDT)

# afisarea mediei acuratetilor din cele 30 de rulari
print(mean(accuraciesSVM))
print(mean(accuraciesDT))

# afisarea deviatiei standard
print(sqrt(var(accuraciesSVM)))
print(sqrt(var(accuraciesDT)))

# afisarea matricei de confuzie
print(contabSVM)
print(contabDT)

# Cohen's k pentru calcului concordantei predictiilor cu iesirile reale ale datelor
Kappa.test(svm.pred, testset[, classColumn])
Kappa.test(rpart.pred, testset[, classColumn])
```

Exemplificare in R 4/4

```
# Curba ROC
# predictie prin probabilitati
# se seteaza probability = TRUE si la antrenarea modelului svm
svm.probabilities <- predict(svm.model, testset[, -classColumn], probability = TRUE)
# se extrage coloana cu probabilitatile pentru a eticheta pozitiv
svm.predictii <- attr(svm.probabilities,"probabilities")[,1]
# se extrag etichetele reale ale datelor
# se convertesc la numere intregi, apoi la 1 pentru pozitiv si 0 pentru negativ
reale <- testset[classColumn][,1]
etichete <- as.integer(reale) - 1
pred <- prediction(as.vector(svm.predictii), etichete)
perf <- performance(pred, "tpr", "fpr")
plot(perf)
aucSVM <- performance(pred, 'auc') # valoarea AUC
print(aucSVM)
# Wilcoxon signed-rank test
wilcox.test(accuraciesSVM, accuraciesDT, paired = TRUE)
```

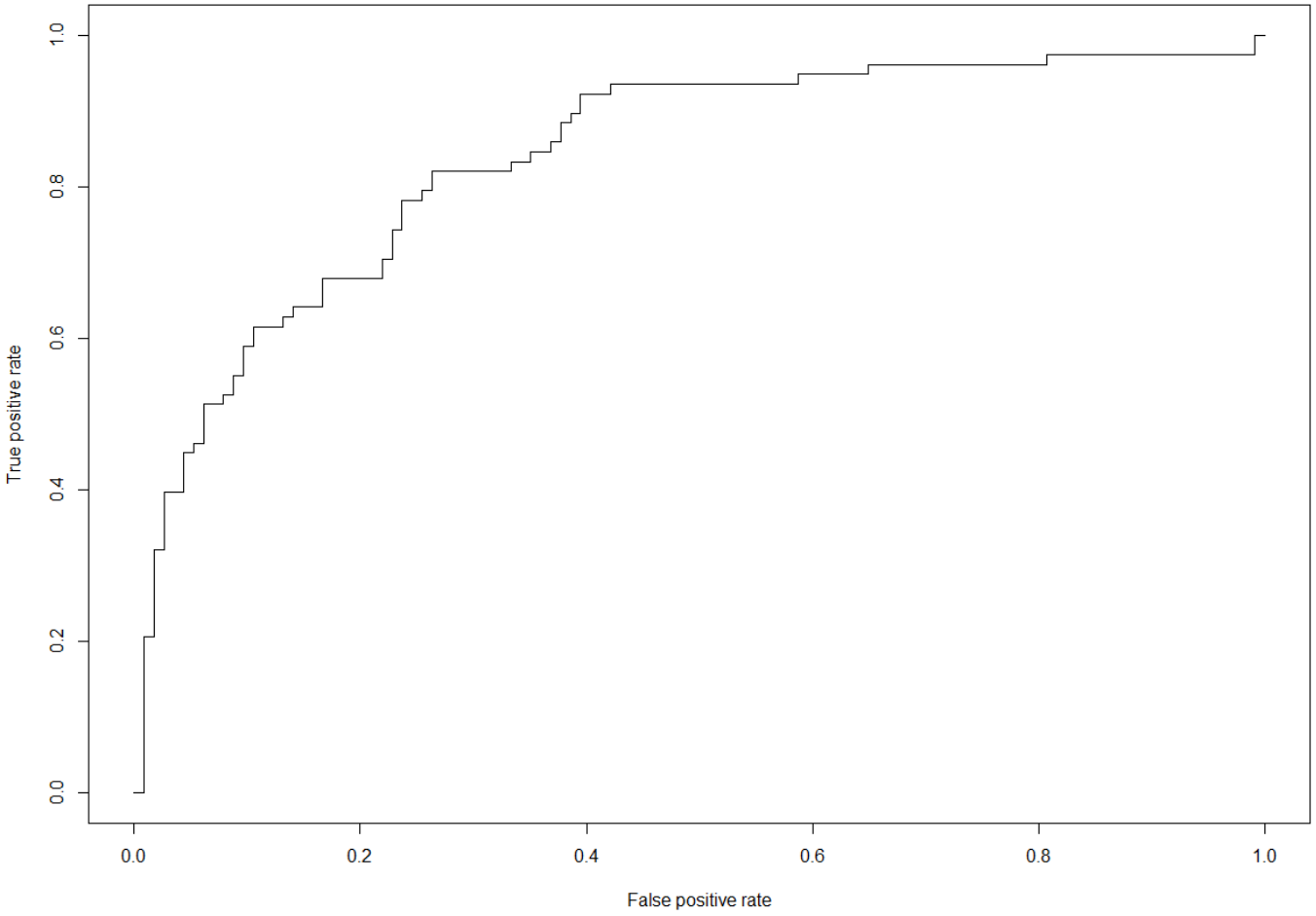
Acuratete medie, deviatie standard, matrice de confuzie

```
> print(accuraciesSVM)
 [1] 0.7552083 0.7760417 0.7864583 0.7708333 0.7291667 0.7552083 0.7708333 0.7604167 0.7864583 0.7968750 0.7552083 0.7864583
[13] 0.7708333 0.7708333 0.8020833 0.7447917 0.8072917 0.7604167 0.7968750 0.7916667 0.8072917 0.7552083 0.8281250 0.7447917
[25] 0.8020833 0.8072917 0.7343750 0.7968750 0.7864583 0.7708333
> print(accuraciesDT)
 [1] 0.7656250 0.7760417 0.7447917 0.7447917 0.6979167 0.7291667 0.7500000 0.7500000 0.7343750 0.7968750 0.7343750 0.6822917
[13] 0.7083333 0.6979167 0.8333333 0.7552083 0.7864583 0.7083333 0.7239583 0.7500000 0.7760417 0.7291667 0.7500000 0.6927083
[25] 0.7552083 0.7760417 0.6927083 0.7239583 0.7864583 0.7447917
>
> # afisarea mediei acuratetilor din cele 30 de rulari
>
> print(mean(accuraciesSVM))
[1] 0.7769097
> print(mean(accuraciesDT))
[1] 0.7432292
>
> # afisarea deviatiei standard
>
> print(sqrt(var(accuraciesSVM)))
[1] 0.02435575
> print(sqrt(var(accuraciesDT)))
[1] 0.03503615
>
> # afisarea matricei de confuzie
> print(contabSVM)
      true
pred   neg pos
  neg  102  32
  pos   12  46
> print(contabDT)
      true
pred   neg pos
  neg   92  27
  pos   22  51
>
```

Cohen's k

```
> # Cohen's k pentru calcului concordantei predictiilor cu iesirile reale ale datelor
> Kappa.test(svm.pred, testset[, classColumn])
$Result

 Estimate Cohen's kappa statistics and test the null hypothesis that the extent of agreement is same as random
 (kappa=0)

data:  svm.pred and testset[, classColumn]
Z = 6.4951, p-value = 4.15e-11
95 percent confidence interval:
 0.3764901 0.6333552
sample estimates:
[1] 0.5049226


$Judgement
[1] "Moderate agreement"

> Kappa.test(rpart.pred, testset[, classColumn])
$Result

 Estimate Cohen's kappa statistics and test the null hypothesis that the extent of agreement is same as random
 (kappa=0)

data:  rpart.pred and testset[, classColumn]
Z = 6.1676, p-value = 3.466e-10
95 percent confidence interval:
 0.3364382 0.5947138
sample estimates:
[1] 0.465576


$Judgement
[1] "Moderate agreement"
```

Curba ROC si valoarea AUC > 0.5




```
> aucSVM <- performance(pred, 'auc')
> print(aucSVM)
An object of class "performance"
Slot "x.name":
[1] "None"

Slot "y.name":
[1] "Area under the ROC curve"

Slot "alpha.name":
[1] "none"

Slot "x.values":
list()

Slot "y.values":
[[1]]
[1] 0.8397436


Slot "alpha.values":
list()
```

Testul Wilcoxon, p-value < 0.01

```
> wilcox.test(accuraciesSVM, accuraciesDT, paired = TRUE)

 Wilcoxon signed rank test with continuity correction

data:  accuraciesSVM and accuraciesDT
V = 361, p-value = 3.723e-05
alternative hypothesis: true location shift is not equal to 0

Warning messages:
1: In wilcox.test.default(accuraciesSVM, accuraciesDT, paired = TRUE) :
  cannot compute exact p-value with ties
2: In wilcox.test.default(accuraciesSVM, accuraciesDT, paired = TRUE) :
  cannot compute exact p-value with zeroes
```

Exercitii 1/2

- Construiti un program R care sa antreneze un arbore de decizie pentru problema diagnozei cancerului de san (Wisconsin breast cancer diagnosis) [1] din pachetul R mlbench [2].
  - Realizati cross-validation prin random subsampling cu 30 de rulari.
  - Obtineti acuratetea medie, deviatia standard si matricea de confuzie.
  - Calculati concordanta cu iesirile adevarate ale datelor cu Cohen's k.

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)
[2] http://cran.r-project.org/web/packages/mlbench/mlbench.pdf

Exercitii 2/2

- Obtineti curba ROC si valoarea AUC.
  - Observatie: pentru a obtine predictie prin probabiliti si nu clase se foloseste type = c(“prob") (in loc de type = c(“class")) ca argument al functiei predict.
- Aplicati testul Wilcoxon pentru a compara predictia arborelui de decizie cu predictia unui clasificator SVM.