Masini cu suport vectorial
Support vector machines (SVM)

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

Bibliografie

- Catalin Stoean, Ruxandra Stoean, Support Vector Machines and Evolutionary Algorithms for Classification: Single or Together?, Intelligent Systems Reference Library, Volume 69, Springer, 2014.

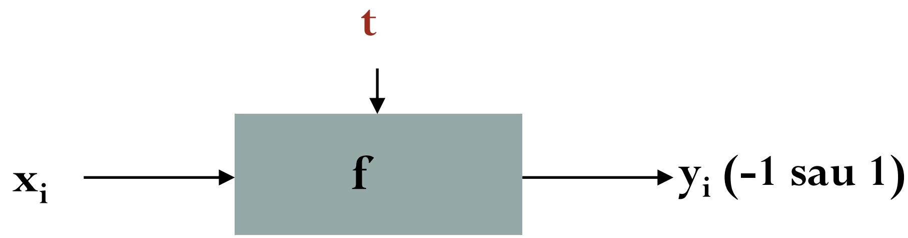
- Simon O. Haykin , Neural Networks and Learning Machines (3rd Edition), Prentice Hall, 2008

Prezentare

- Unelte foarte puternice in:
  - Clasificare
    - Recunoasterea scrisului
    - Recunoasterea obiectelor
    - Recunoasterea vorbirii
    - Categorizare de texte
    - In principiu binara – metode de extinder… multe clase
  - Regresie
- Un tip de masina instruibila supervizata

• **Faza de antrenament:** invatare *(exemplu → eticheta)*
• **Faza de test**: predictie asupra *(exemplu nou → ?)*

Clasificare binara



- O multime de date de antrenament

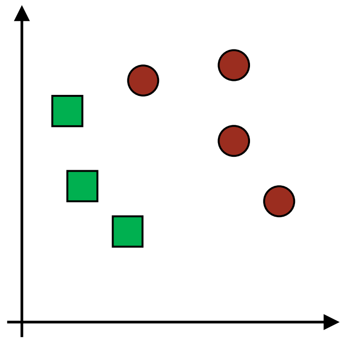
$$\{(x_i, y_i)\}_{i=1,2,\ldots,m} \qquad x_i \in R^n \qquad y_i \in \{-1,+1\}$$

- O familie de functii

$$\{f_t \mid t \in T\} \qquad f_t : \Re^n \to \{-1,+1\}$$

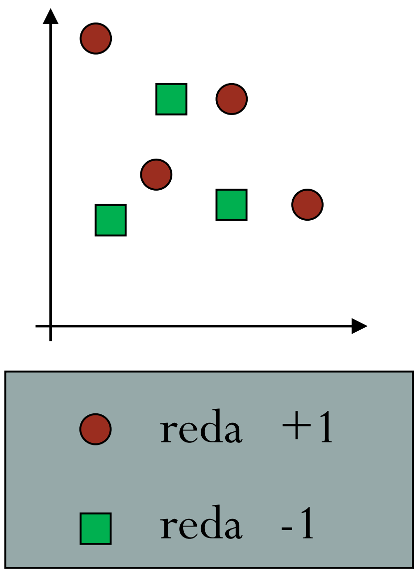
- Gaseste parametrii t pentru a invata corespondenta optima intre fiecare x si y

Invatarea in cadrul SVM



Liniar separabile

Masini cu suport vectorial liniare

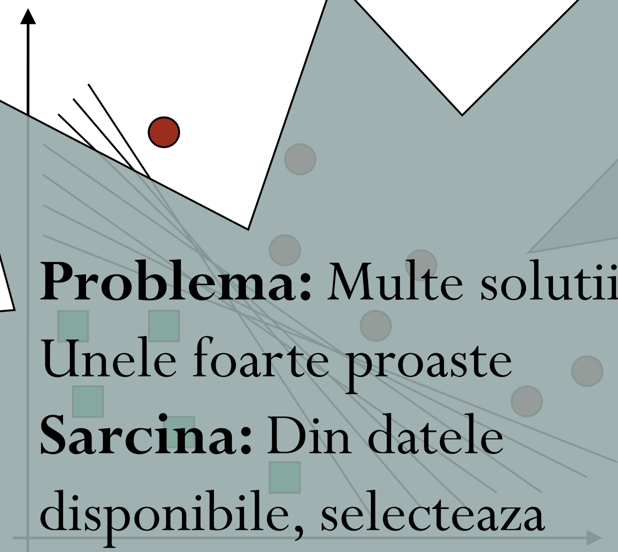


Liniar neseparabile

Masini cu suport vectorial liniare

Masini cu suport vectorial neliniare

SVM liniare pentru date separabile



Cum se pot clasifica aceste date?

Oricare dintre acestea ar fi bune...

Dar care este cel mai bun?

Problema: Multe solutii!

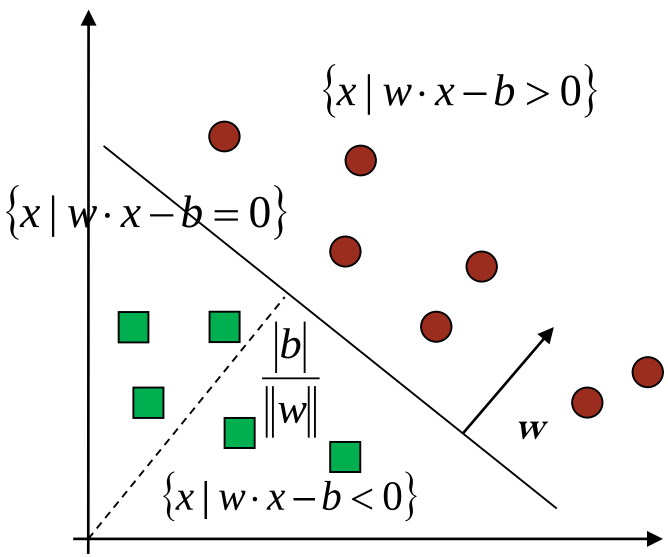
Unele foarte proaste

Sarcina: Din datele disponibile, selecteaza hiperplanul care separa bine “in general”

reda +1

reda -1

SVM liniare pentru date separabile



Existenta unui hiperplan de separatie cu ecuatia: $\langle w, x \rangle - b = 0$

Doua submultimi sunt liniar separabile daca si numai daca exista $w \in R^n$ si $b \in R$ a.i.:

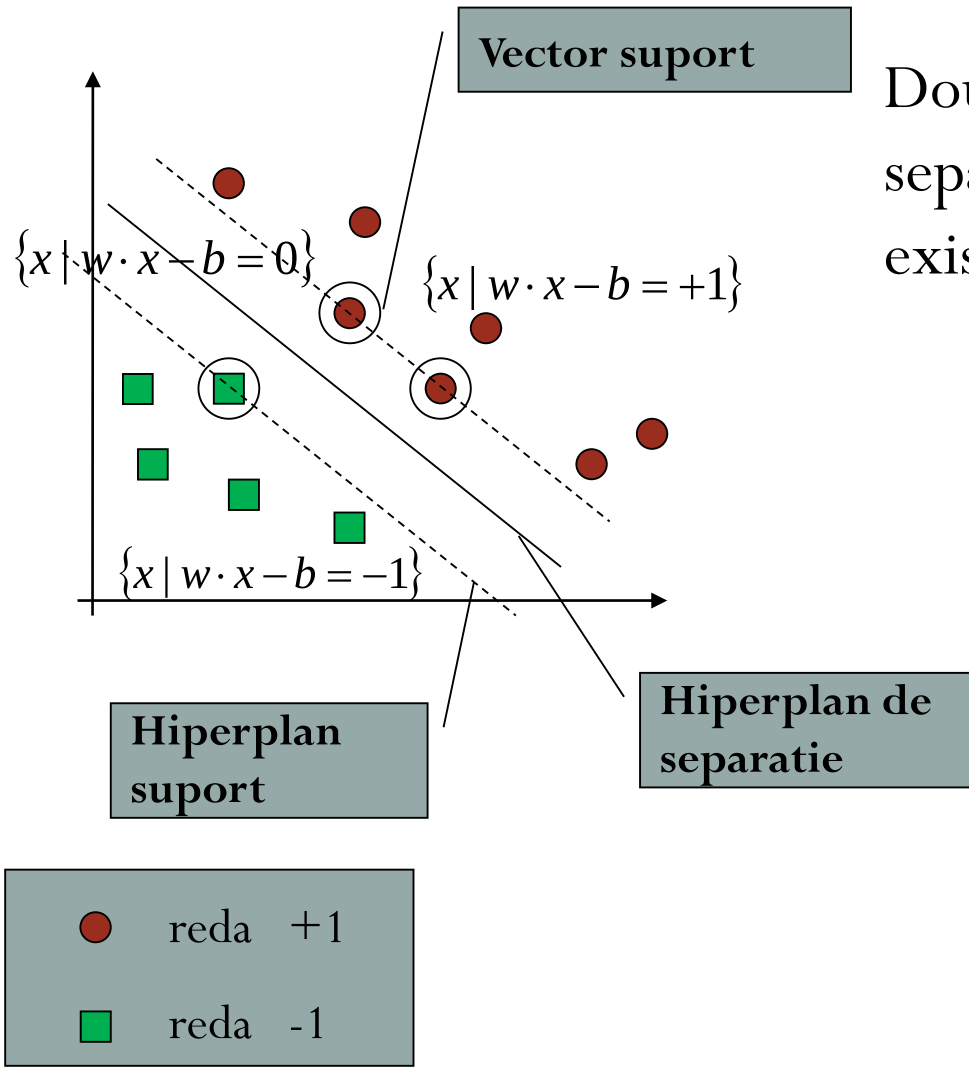
$$\begin{cases} w \cdot x_i - b \geq 0, \ y_i = +1 \\ w \cdot x_i - b \leq 0, \ y_i = -1 \end{cases}$$

$$i = 1,2,\ldots,m$$

- reda +1
- reda -1

- w – vectorul ponderilor
- b – bias (deplasare)

SVM liniare pentru date separabile



Doua submultimi sunt liniar separabile daca si numai daca exista $w \in R^n$ si $b \in R$ a.i.:

$$\begin{cases} w \cdot x_i - b \geq +1, y_i = +1 \\ w \cdot x_i - b \leq -1, y_i = -1 \end{cases}$$

$$i = 1,2,\ldots,m$$

SVM liniare pentru date separabile

- O familie de functii:

$$\left\{ f_{w,b} \mid w \in \Re^n, b \in \Re \right\} \qquad f_{w,b} : \Re^n \to \{-1, 1\} \qquad f_{w,b}(x) = \operatorname{sgn}(w \cdot x - b)$$

- O multime de date de antrenament:

$$\{(x_i, y_i)\}_{i=1,2,\ldots,m} \qquad x_i \in R^n \qquad y_i \in \{-1, +1\}$$

- Invata corespondenta optima

Principiul minimizarii riscului

- Pentru o sarcina de invatare, cu o anumita cantitate finita de date de antrenament, o masina este un bun generalizator daca:
  - Are o acuratete buna pe multimea de antrenament data
  - Are capacitatea de a invata si alte multimi de antrenament fara eroare

Hiperplanul optim la SVM liniare pentru date separabile

$w = ?$ **si** $b = ?$ ***a.i.***

1. **Acuratete** $y_i(w \cdot x_i - b) - 1 \geq 0 \quad i = 1,2,\ldots,m$

2. **Marginea de separatie maxima**

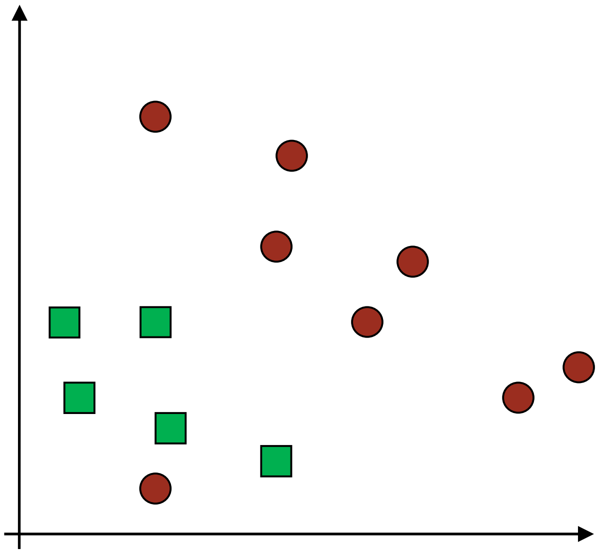
Latimea cu care granita dintre clase poate fi marita pana la a lovi un exemplu

$\Rightarrow$ **minimizeaza** $\dfrac{\|w\|^2}{2}$ **De ce?**

Distanta de la fiecare cel mai apropiat punct de hiperplanul de separatie din cele doua parti ale sale este :

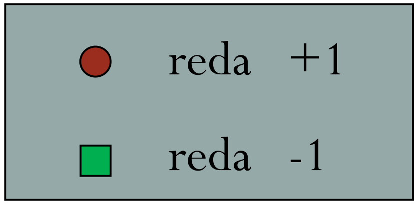
$$\frac{|w \cdot x_i - b|}{\|w\|} = \frac{1}{\|w\|} \quad \Rightarrow \frac{2}{\|w\|} \longrightarrow \textbf{maxim}$$

SVM liniare pentru date neseparabile

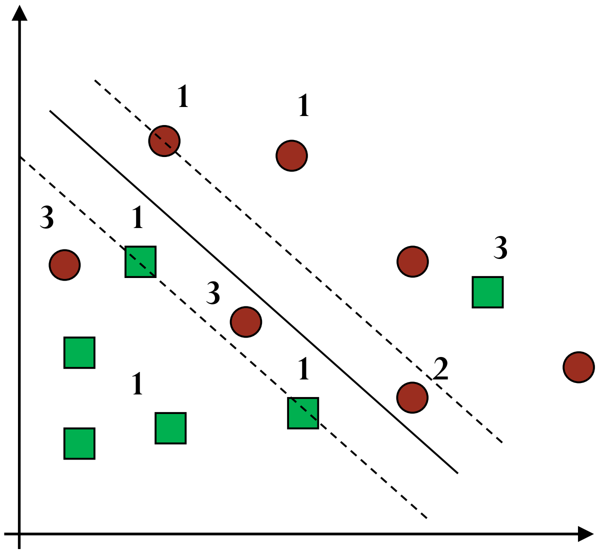


Cum separam aceste noi date?

Putem relaxa conditia de separare.



SVM liniare pentru date neseparabile



Fiecare vector de antrenament are o deviatie de la hiperplanul sau suport de

$$\pm \frac{\xi_i}{\|w\|} \qquad \xi_i \geq 0$$

$\xi_i = 0$	**1 Corect**
$\xi_i < 1$	**2 Corect (in margine)**
$\xi_i > 1$	**3 Eroare**

●	reda $+1$
■	reda -1

Hiperplanul optim la SVM liniare pentru date neseparabile

- Se relaxeaza constrangerile :

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i$$

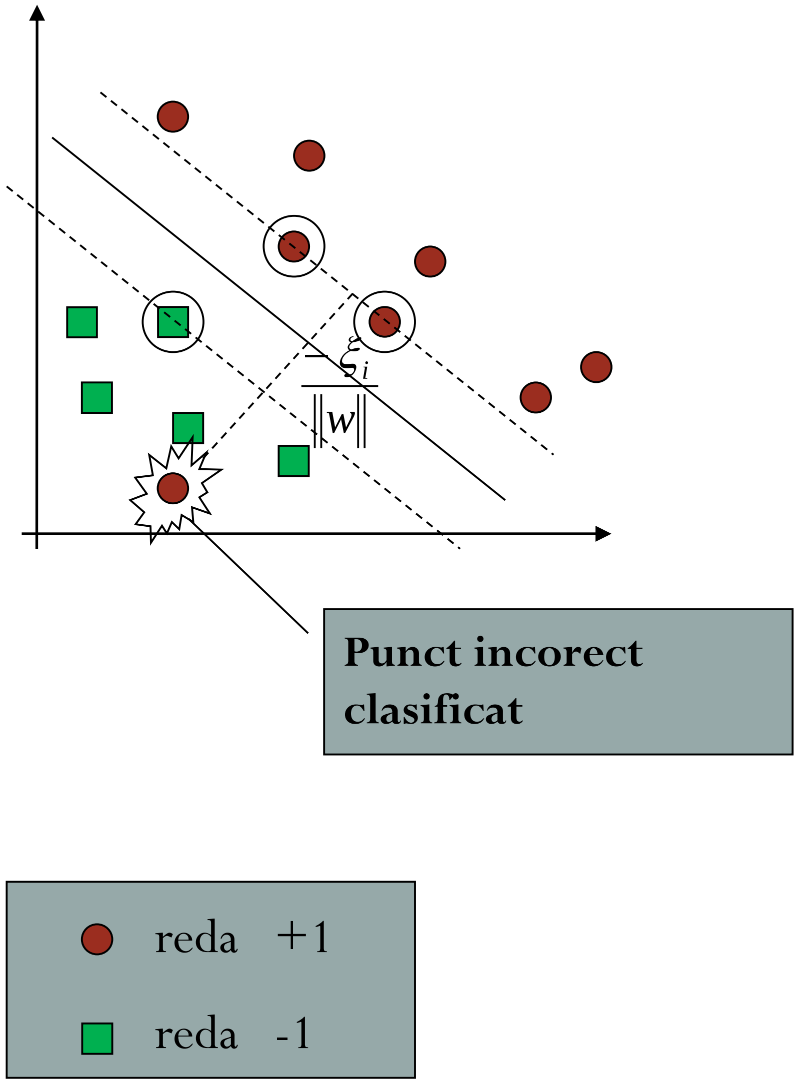
$$\xi_i \geq 0 \quad i = 1,2,\ldots,m$$

- Suma clasificarilor gresite trebuie minimizata impreuna cu maximinarea marginii de separatie:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$C > 0$$

SVM liniare pentru date neseparabile

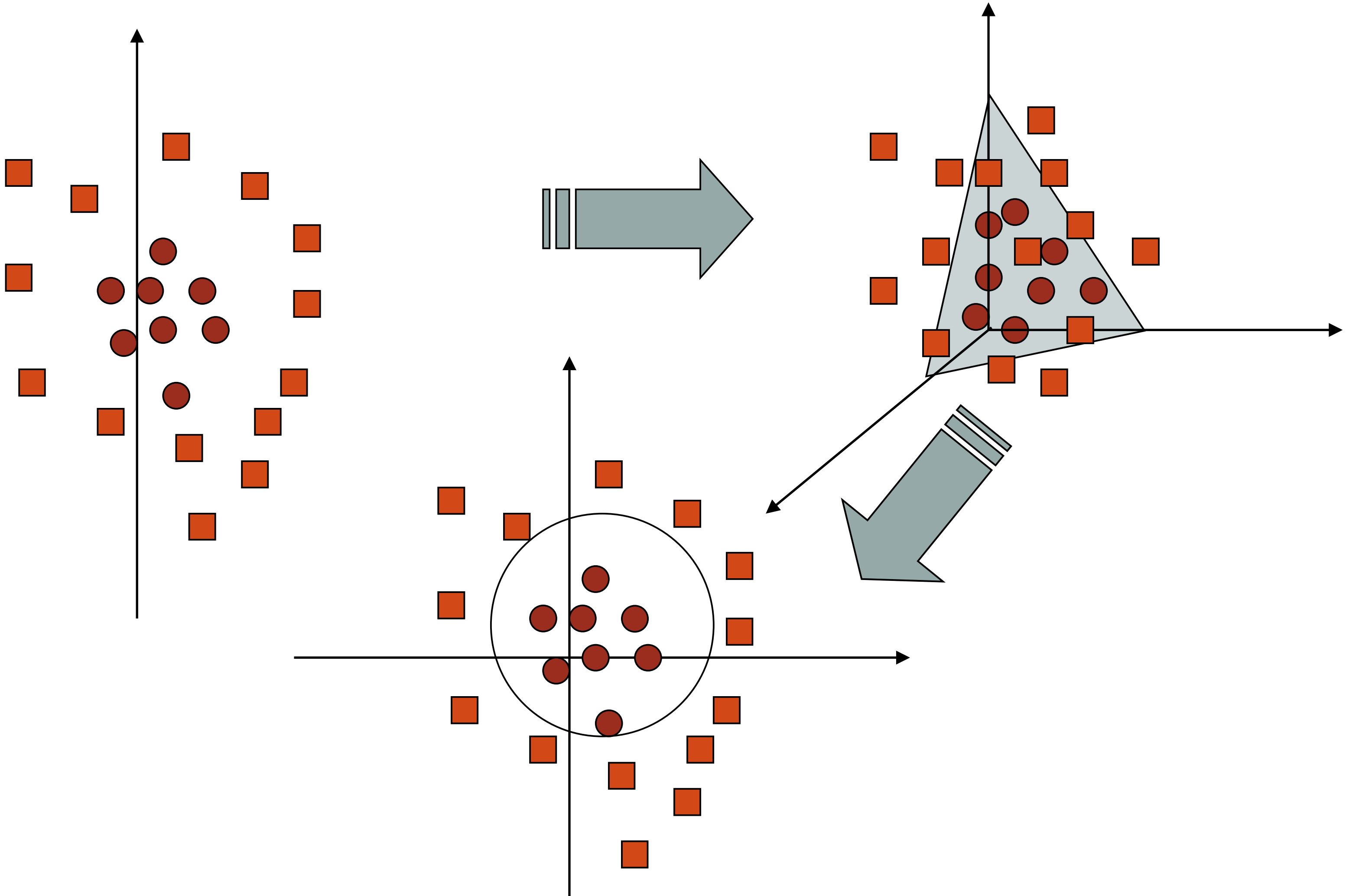


SVM neliniare

- Un spatiu de ... are poa ... at in ... un spatiu nou al trasaturilor u...
  - transf...
  - din...

Se pot generaliza conceptele pentru a construi un hiperplan care sa nu fie o functie liniara?

SVM neliniare



Transformare neliniara

x - exemplu din spatiul de intrare

$\Phi : \Re^n \to H$ - transformarea

Ar putea fi posibila existenta unei functii kernel *K*…

$$K(x, y) = \Phi(x) \cdot \Phi(y)$$

- Se transforma problema de optimizare in formulare duala.

Cum va fi perceputa transformarea la nivelul problemei de optimizare?

- In expresia duala, datele apar numai ca parte a unor produse scalare.
- In expresia determinarii hiperplanului optim in H, datele vor aparea, de asemenea, ca parte a unor produse scalare.

Transformarea neliniara

- Kernel – functie care exprima un produs scalar intr-un spatiu al trasaturilor.

- Convergenta masinilor cu suport vectorial catre o solutie impune ca o astfel de functie sa fie pozitiv (semi-)definita.
- Un astfel de kernel este unul care satisface teorema lui Mercer.
- Face imposibila detectarea de suprafete eficiente dar care nu o respecta.

Kernel

- Kernel – functie care exprima un produs scalar intr-un spatiu al trasaturilor.
- Exista kernele clasice pentru care s-a demonstrat respectarea conditiei lui Mercer:
  - Kernelul polinomial: $K(x,y) = \left(x \cdot y\right)^p$
  - Kernelul radial: $K(x,y) = e^{\frac{\|x-y\|^2}{\sigma}}$

Rezolvarea problemei de optimizare

- Se bazeaza pe notiuni de convexitate.
- Se construieste Lagrangianul si se aplica conditiile Karush-Kuhn-Tucker-Lagrange (KKTL).
- Se construieste duala problemei de optimizare.
- Se egaleaza gradientul noii functii obiectiv cu 0 si se rezolva sistemul rezultat.
- ***w*** si ***b*** rezulta din conditiile KKTL, daca se pot calcula;
  - daca nu, masina cu suport vectorial va determina direct clasa pentru vectorii de test.
- Semnul lui $f_{w,b}$ da clasa finala – pozitiva sau negativa.

SVM pentru mai multe clase

- In utilizarea standard, SVM este o masina instruibila pentru probleme binare (cu doua clase).

- In extinderea la mai multe clase (*k*), exista doua tehnici clasice:

  - One-against-all (Unul impotriva tuturor)

  - One-against-one (Unu la unu)

One-against-all (1aa)

- Se construiesc k clasificatori SVM.
  - Pentru al i-lea SVM, clasa i este pozitiva si celelalte impreuna reprezinta clasa negativa.

- Fiecare al i-lea SVM determina hiperplanul optim de coeficienti w^i si b^i.

- Eticheta pentru un exemplu nou este data de clasa care are valoarea maxima pentru functia invatata f_{w^i,b^i}.

One-against-one (1a1)

- Se construiesc k(k-1)/2 clasificatoare SVM pentru fiecare doua clase pe rand.
  - Clasa i este clasa pozitiva, iar j cea negativa

- Se determina hiperplanul de decizie intre fiecare doua clase i si j cu coeficientii w^{ij} si b^{ij}.

- Eticheta pentru un exemplu nou se determina prin vot.
  - Pentru fiecare SVM se calculeaza clasa exemplului, care primeste un vot.
  - Castiga clasa cu cele mai multe voturi obtinute.

SVM pentru regresie

- In acest caz, datele sunt constranse sa se afle pe un hiperplan care
  - Permite o anumita eroare ϵ fata de iesirile reale
  - Are abilitate mare de generalizare

- Pentru cazul liniar:

$$\begin{cases} y_i - w \cdot x_i + b \le \varepsilon \\ w \cdot x_i - b - y_i \le \varepsilon \end{cases} \qquad i = 1,2,\ldots,m$$

  - si minimizeaza $\|w\|^2$

Problema de optimizare - regresie

- Pentru cazul general al datelor neseparabile:

$$\begin{cases} y_i - w \cdot x_i + b \leq \varepsilon + \xi_i \\ w \cdot x_i - b - y_i \leq \varepsilon + \xi_i^* \end{cases} \qquad \xi_i, \xi_i^* \geq 0$$

  - si minimizeaza $\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \qquad C > 0$

- Aplicarea de kernele este similara cu cea in cazul clasificarii.
- Functia $f_{w,b}$ rezultata da iesirea prognozata pentru un nou exemplu.

SVM in R. Clasificare (1/4)

```
library(e1071) # pentru SVM
library(mlbench) #pentru baza de date cu diabet
data(PimaIndiansDiabetes)
dat <- PimaIndiansDiabetes

# De 30 de ori vom partitiona aleator baza in antrenament si test pentru validare incrucisata (cross-validation)
repeats <- 30
classColumn <- 9
accuracies <- vector(mode="numeric",length=10) # cele 30 de acurateti obtinute
```

SVM in R. Clasificare (2/4)

```
index <- 1:nrow(dat)
for(i in 1:repeats){
        testindex <- sample(index, trunc(length(index)/4)) # 25% date in test/75% in antrenament
        testset <- dat[testindex, ]
        trainset <- dat[-testindex, ]

        svm.model <- svm(diabetes ~ ., data = trainset, kernel = "linear", cost = 1)
        svm.pred <- predict(svm.model, testset[, -classColumn])
        contab <- table(pred = svm.pred, true = testset[, classColumn])
        accuracies[i] <- classAgreement(contab)$diag
}
```

SVM in R. Clasificare (3/4)

```
print(accuracies)
print(mean(accuracies)) # media acuratetilor
print(sqrt(var(accuracies))) # deviatia standard
print(summary(accuracies))

print("Un exemplu de matrice de confuzie")
print(contab) # cate date din fiecare clasa au fost prognozate in clasa respectiva, respectiv duse in clasa opusa
```

SVM in R. Clasificare (4/4)

```
> print(accuracies)
 [1] 0.7708333 0.7864583 0.7031250 0.8177083 0.7708333
 [6] 0.7812500 0.7812500 0.7239583 0.7760417 0.7500000
[11] 0.7812500 0.8125000 0.8020833 0.7500000 0.7031250
[16] 0.7604167 0.7864583 0.8385417 0.7552083 0.8072917
[21] 0.7968750 0.7864583 0.7083333 0.8020833 0.7656250
[26] 0.7395833 0.7604167 0.7760417 0.7916667 0.7812500
> print(mean(accuracies))
[1] 0.7722222
> print(sqrt(var(accuracies)))
[1] 0.0330519
> print(summary(accuracies))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.7031  0.7565  0.7786  0.7722  0.7904  0.8385
>
> print("Un exemplu de matrice de confuzie")
[1] "Un exemplu de matrice de confuzie"
> print(contab)
     true
pred  neg pos
  neg 117  29
  pos  13  33
```

- Rezultatul programului:
  - Acuratetile in cele 30 de rulari.
  - Acuratetea in medie.
  - Deviatia standard.
  - Sumarul privind acuratetea in cele 30 de rulari.
  - Matricea de confuzie:
    - Pe linie – predictia
    - Pe coloana - realitatea

SVM in R. Regresie (1/4)

```
library(e1071)
library(mlbench)

data(BostonHousing)

classColumn <- 14

# o singulara rulare cu multimile de antrenament si test alese aleator
random <- sample(1:506, replace = FALSE)
randomTrain <- random[1:380]
randomTest <- random[381:506]
train <- BostonHousing[randomTrain,]
test <- BostonHousing[randomTest,]
```

SVM in R. Regresie (2/4)

```
svm.model <- svm(medv ~ ., data = train, kernel = "radial",
epsilon = 0)
svm.pred <- predict(svm.model, test[, -classColumn])
summary(svm.pred)

# Eroarea RMS (root mean square error - RMSE) pentru n date
cu valorile prognozate Pi si cele reale Ri.
```

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - R_i)^2}$$

```
RMSE <- sqrt(mean((svm.pred - test[, classColumn])^2))
print(RMSE)
```

SVM in R. Regresie (3/4)

#calculeaza corelatia dintre prognozat si real

Coeficientul de corelatie al lui Pearson masoara gradul de potrivire intre iesirile modelului si cele reale.

#Coeficientul de corelatie bazat pe rang al lui Spearman este calculat drept cel al lui Pearson pentru variabilele ierarhizate.

```
cor.test(svm.pred, test[, classColumn], method = "spearman")
cor.test(svm.pred, test[, classColumn], method = "pearson")
```

SVM in R. Regresie (4/4)

```
> summary(svm.pred)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  8.552  17.270  21.730  23.120  27.840  47.730
>
> RMSE <- sqrt(mean((svm.pred - test[, classColumn])^2))
> print(RMSE)
[1] 4.565229
>
> #calculeaza corelatia dintre prognozat si real
>
> cor.test(svm.pred, test[, classColumn], method = "spearman")

 Spearman's rank correlation rho

data:  svm.pred and test[, classColumn]
S = 19023.73, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
     rho
0.942936

Warning message:
In cor.test.default(svm.pred, test[, classColumn], method = "spearman") :
  Cannot compute exact p-values with ties
> cor.test(svm.pred, test[, classColumn], method = "pearson")

 Pearson's product-moment correlation

data:  svm.pred and test[, classColumn]
t = 21.9085, df = 124, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8489145 0.9225224
sample estimates:
     cor
0.891457
```

- Rezultatul programului:
  - Eroare RMS mica.
  - Testele Pearson si Spearman indica potrivire aproape perfecta (aproape de 1) intre valorile reale si cele prognozate de model cu valoarea p de ordinul e-16.

Exercitii

- Implementati in R un model de clasificator SVM pentru problema diagnozei cancerului de san (Wisconsin breast cancer diagnosis) [1] din pachetul R mlbench [2].

- Implementati in R un model SVM de regresie pentru problema bacsisului din cursul anterior.

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