Support Vector Machine

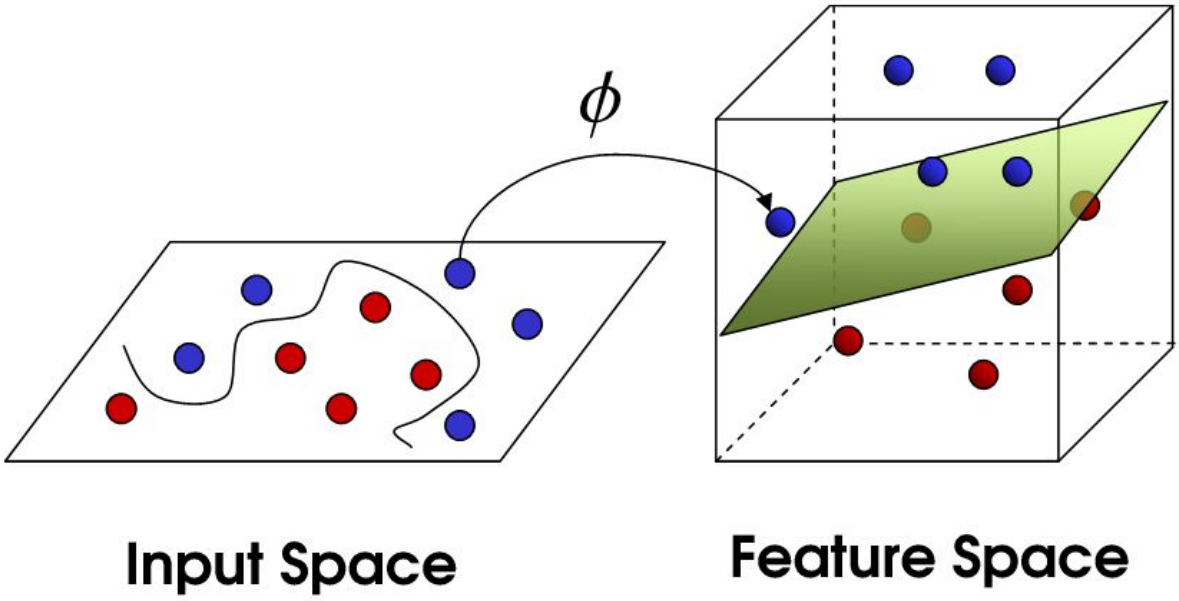
W czasie tego ćwiczenia omówiony zostanie klasyfikator Maszyna Wektorów Podpierających (ang. Support Vector Machine, SVM). Ćwiczenia praktyczne zrealizowane zostaną w oparciu o pakiet statystyczny R.

Zadanie 1

… wprowadzenie

W ramach poznania klasyfikatora zapoznaj się z dokumentem svm.pdf załączonym w pliku z ćwiczeniami.

a) Zastanów się jaki wpływ ma stała C na wyniki klasyfikatora?
b) Przyjmijmy, że mamy zbiór treningowy o rozmiarze m x p (m liczba przypadków treningowych, p liczba cech). W ilu wymiarowej przestrzeni będą realizowane obliczenia dla jąda RBF?



c) Czy SVM jest klasyfikatorem liniowym? Odpowiadają na pytanie posłuż się powyższym rysunkiem.

Autor: Kamil Żbikowski, kzbikows@gmail.com

Zadanie 2

Aby wyrobić intuicję dotyczącą klasyfikatora SVM posłużymy się w niniejszym ćwiczeniu przykładem ze zbioru “cats”.

Najpierw ładujemy zbiór:

```
> data(cats, package = "MASS")
```

W ramach niniejszego ćwiczenia korzystać będziemy z pakietu e1071.

Bibliotekę ładujemy:

```
> library(e1071)
```

lub jeśli nie mamy jej zainstalowanej wykonujemy uprzednio polecenie

```
> install.packages(“e1071”)
```

Analogiczne kroki wykonujemy dla biblioteki “caret”, “ROCR” oraz “lattice”, które będziemy wykorzystywać w dalszej części ćwiczenia.

W kolejnych krokach zbieramy informację o zbiorze wykonując następujące polecenia:

```
> summary(cats)
> dim(cats)
> pairs(cats)
```

a) Jakiego rodzaju zmienne występują w zbiorze?
b) Jakie oceniasz możliwość skutecznej klasyfikację płci na podstawie dwóch pozostałych zmiennych?

Autor: Kamil Żbikowski, kzbikows@gmail.com

Zadanie 3

W tym kroku dla zbioru “cats” sprawdź jak zachowa się prosty klasyfikator oparty o regresję logistyczną i narysuj granicę decyzyjną.

```
> mdl = glm(Sex~., data = cats, family = binomial)
> mdlP = predict(mdl, cats)
> slope <- coef(mdl)[3]/(-coef(mdl)[2])
> intercept <- coef(mdl)[1]/(-coef(mdl)[2])
> xyplot( Bwt ~ Hwt , data = cats, groups = cats$Sex,
        panel=function(...){
          panel.xyplot(...)
          panel.abline(intercept , slope)
          panel.grid(...)
        })
```

Zadanie 4

W kolejnym kroku zastosuj funkcję svm() z pakietu e1071 w celu skonstruowania modelu decyzyjnego w oparciu o klasyfikator SVM. Najpierw zapoznaj się z dokumentacją dotyczącą svm():

```
> ?svm
```

a następnie zbuduj model:

```
> m1 <- svm(Sex~., data = cats, kernel="linear", cost=1)
> plot(m1, cats)
```

Porównaj wyniki z wynikami z Zadania 3.

Zadanie 5

Przeprowadź eksperyment z różnymi wartościami stałej C w klasyfikatorze SVM. Sprawdź wartości {1,50, 100, 10e6}. Prównaj wyniki dla różnych wartości C. Następnie zastosuj jako jądro radialną funkcję bazową. Przeprowadź ekseperyment dla różnych wartości C z jądrem RBF. Co zaobserwowałeś?

Autor: Kamil Żbikowski, kzbikows@gmail.com

Dla poszczególnych wartości parametru C przeprowadź predykcję na całym zbiorze za pomocą funkcji predict(). Następnie z pomocą funkcji table() oraz funkcji confusionMatrix() z pakietu caret oblicz macierz pomyłek.
Funkcja confusionMatrix() zwraca szereg interesujących statystyk. Przenalizuj następujące kwestie:

a) Jak zmienia się wartość Accuracy w zależności od zmian stałej C?
b) Jak zmienia się wartość Balanced Accuracy? Jak zdefiniowana jest ta statystyka?
c) Co mówią nam wartości statystyk Sensitivity, Specificity, Negative Prediction Value, Positive Prediction Value?
d) Zastanów się nad przydatnością statystyk z c) w kontekście klasyfikacji binarnej np. detekcji zdarzeń dotyczących niespłacalności kredytów lub wykrycia nowotworu? Które ze statystyk chcielibyśmy optymalizować?

Zadanie 6

Policz liczności dla poszczególnych klas ze zbioru Cats. Zapisz w zmiennej numberOfF liczbę przykładów ze zmienną Sex=F a w zmiennej numberOfM liczbę przykładów ze zmienną Sex=M. Co możemy powiedzieć o rozkładzie tych klas w badanej populacji?

Zapoznaj się z parametrem class.weights w klasyfikatorze svm. Jaka jest jego rola?

Wprowadź wagi dla parametru class.weights zgodnie obliczone w następujący sposób:

```
> wF=1
> wM=numberOfF/numberOfM
```

Jakie zmiany w statyskach klasfyfikatora zaobserwowałeś?
Zwiększ o 0.1 wartość parametru wM. Jaki wpływ na Accuracy klasyfikatora ma ta zmiana?

Zadanie 7

Jądro RBF jako możemy parametryzować podając do funkcji svm() parametr “gamma”. Sprawdź różne wartości parametru {0.1, 1, 5, ,50, 500}. Zwróć uwagę na statystyki klasyfikacji zwracane przez funkcję confusionMatrix(). Którą z wartości parametru gamma wybrałbyś dla docelowego modelu? Uzasadnij swój wybór.

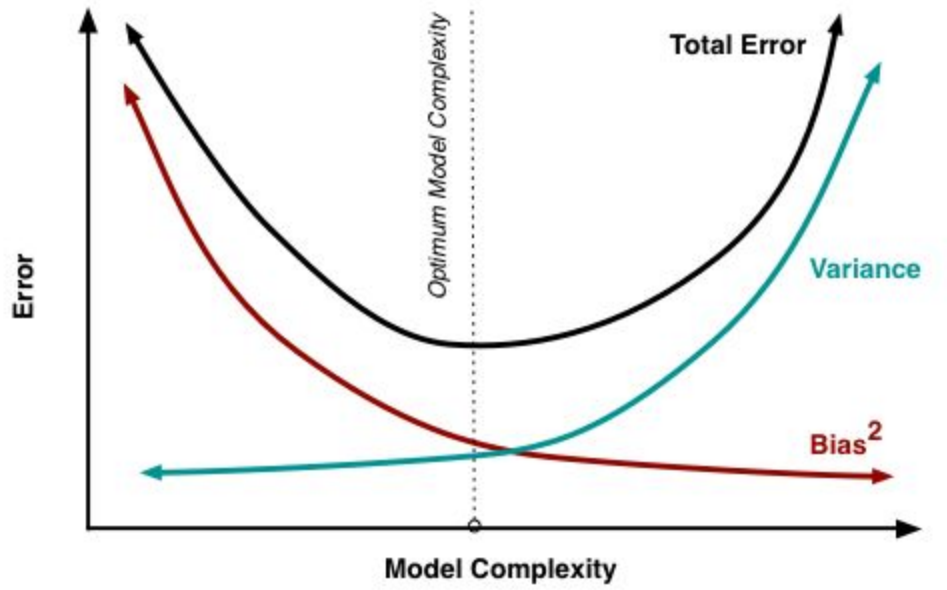
Autor: Kamil Żbikowski, kzbikows@gmail.com

Zadanie 9

Podziel zbiór cats na dwie częsci próbę trenującą oraz testową w proporcji 3:7

```
> nmbOfCats = dim(cats)[1]
> trIndx = sample(nmbOfCats, nmbOfCats*0.7)
> catsTr = cats[trIndx, ]
> catsTst = cats[-trIndx, ]
```

a) Dla zbioru catsTr skonstruuj model z wybranym parametrem gamma z Zadania 8., który dawał najlepsze rezeultaty.
b) Następnie wykonaj predykcję dla zbioru catsTst modelem z punktu a) Co zaobserwowałeś w zakresie wartości statystyk klasyfikacji? Z czego wyniakają takie a nie inne wartości?
c) Zinterpretuj poniższy wykres.



Autor: Kamil Żbikowski, kzbikows@gmail.com

Zadanie 10

"**K-fold cross validation** is one way to improve over the holdout method. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times. The variance of the resulting estimate is reduced as k is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set k different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over."

W ramach pakietu e1071 zapoznaj się z funkcją tune(), dzięki której możliwe jest wykonanie walidacji krzyżowej.
Podaj kilka wartości z parametrów gamma, C oraz class.weights na wejście funkcji tune(). Uruchom walidację krzyżową. Skomentuj otrzymane wyniki.

Zadanie 11

Zadanie to będzie łączyło wiedzę z poprzednich zadań. Dane wykorzystywane w ramach tego zadania dotyczą zdarzeń tzw. "defaultów" kredytowych czyli nie danych dotyczących niespłacenia kredytów.
Dane załaduj w następujący sposób:

```
> def = read.csv("./defaults.csv", sep = ";", header = TRUE)
```

W trakcie wstępnej analizy przydatne jest pracowanie z mniejszą próbką danych. W tym celu ogranicz się do 2000 losowo wybranych przykładów:

```
> smpl = def[sample(nrow(def),2000),]
```

….. skontruj optymalny model, który zachowa właściwości generalizacji. Pod koniec zajęć otrzymasz zestaw danych walidacyjnych. Zostanie wyłoniony najlepszy klasyfikator….

W ramach budowy kolejnych modeli w tym zadaniu posłuż się wykreślaniem krzywej ROC, którą (zakładając, że model nazywa się svm.fit) zbudujesz w następujący sposób:

```
>fitted=attributes(predict(svm.fit,smpl,decision.values=T))$decisio
n.values
> predob =  prediction(fitted, smpl$default.payment.next.month)
> perf = performance(predob, 'tpr', 'fpr')
> plot(perf)
```

Do oceny jakości klasyfikacji dla udostępnionego pliku użyć wartość “balanced accuracy” ze zmiennej cm:

```
def_w = read.csv("./defaults_valid.csv", sep = ";", header = TRUE)
def_w[,25] = as.factor(def_w[,25])
defaultsResults = predict(svm.fit, def_w)
ctab = table(defaultsResults, def_w$default.payment.next.month)
cm = confusionMatrix(ctab)
```