

Prácticas

Sistemas Inteligentes I

Sesión 9. Máquinas Vectores Soporte

José A. Montenegro Montes

monte@lcc.uma.es

Resumen

- Práctica Máquinas de Vectores Soporte

AIMA

Práctica

Máquinas de Vectores Soporte



Máquinas de Vectores Soporte

Diseñar 3 máquinas de vectores soporte (MVS) para cada uno de los datos administrados (Iris, Glass y Satimage) con la mejor clasificación posible.

Para ello se deberá utilizar la aplicación `intsys.svm.SVM_App.java` y guardar cada uno de los 3 modelos.

Iris	
Clases	3
Patrones	150
Atributos	4

Glass	
Clases	6
Patrones	214
Atributos	9

Satimage	
Clases	6
Patrones	4435
Atributos	36

http://en.wikipedia.org/wiki/Iris_flower_data_set

<http://archive.ics.uci.edu/ml/datasets/Iris>

<http://archive.ics.uci.edu/ml/support/Glass+Identification>

[https://archive.ics.uci.edu/ml/datasets/Statlog+\(Landsat+Satellite\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite))

Máquinas de Vectores Soporte

- a) Las definiciones de las “kernel function” están en el método `kernel_function` de la Clase `Kernel` en el archivo `svm.java`. En la definición de las funciones están los significados de sus parámetros.
- b) Seleccionar en cada problema (Iris, Glass o Satimage) la mejor configuración (kernel function y parámetros), es decir, aquellas que obtengan un valor más alto de la variable accuracy. Almacenar cada solución con la opción `save` del menú de la izquierda.
- c) Subir un zip con los tres archivos de la mejor configuración para cada problema: `iris.svm`, `glass.svm`, `satimage.svm`

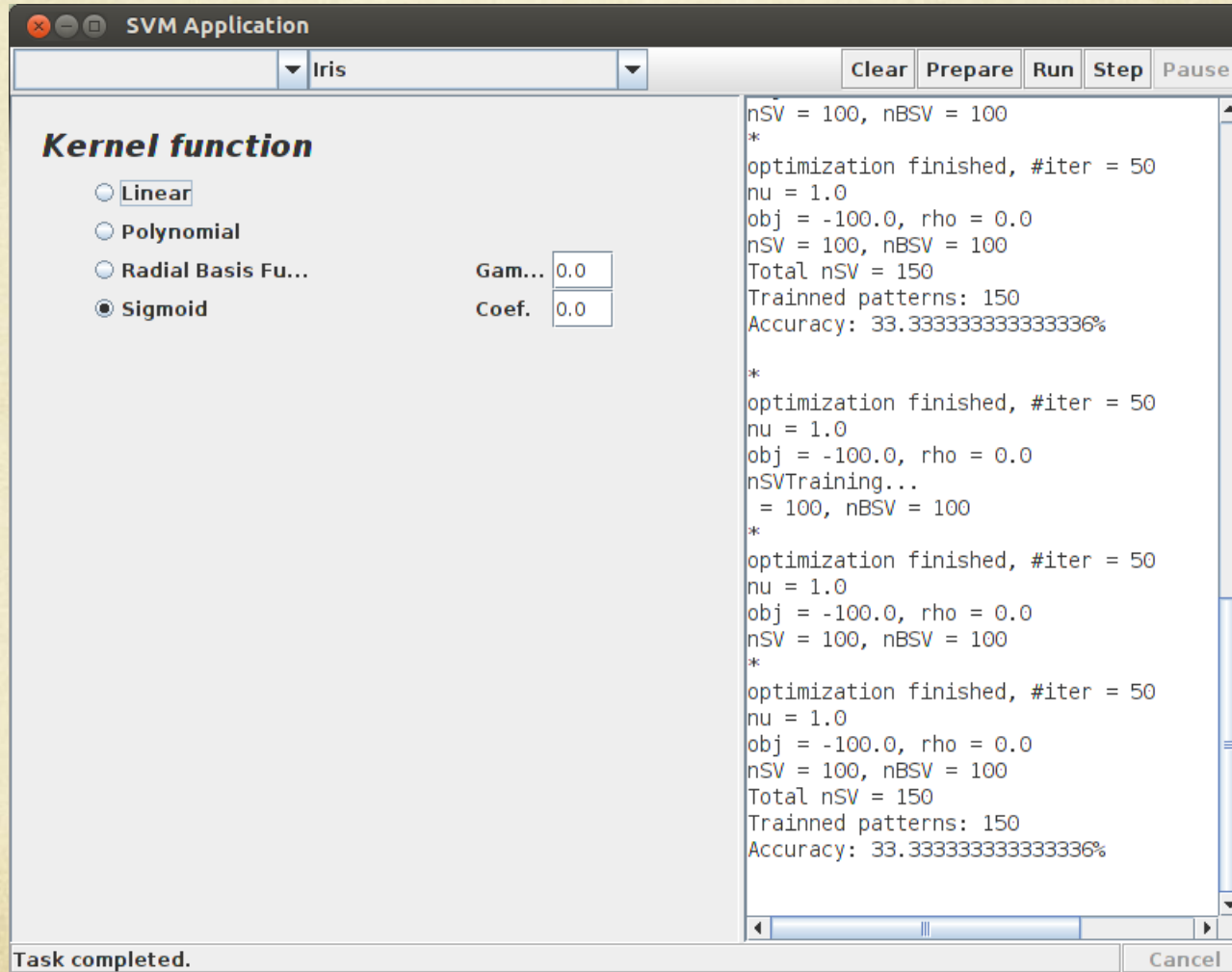
Máquinas de Vectores Soporte

kernel_function

```
case svm_parameter.LINEAR:  
    return dot(x[i],x[j]);  
case svm_parameter.POLY:  
    return powi(gamma*dot(x[i],x[j])+coef0,degree);  
case svm_parameter.RBF:  
    return Math.exp(-gamma*(x_square[i]+x_square[j]-2*dot(x[i],x[j])));  
case svm_parameter.SIGMOID:  
    return Math.tanh(gamma*dot(x[i],x[j])+coef0);
```

linear:	$u \cdot v$
polynomial:	$(\gamma u \cdot v + \text{coef0})^{\text{degree}}$
radial basis function:	$\exp(-\gamma u-v ^2)$
sigmoid:	$\tanh(\gamma u \cdot v + \text{coef0})$

Máquinas de Vectores Soporte



The screenshot shows a software application window titled "SVM Application". The dataset is set to "Iris". The "Kernel function" section has four radio buttons: "Linear", "Polynomial", "Radial Basis Fu...", and "Sigmoid", with "Sigmoid" selected. To the right, there are two input fields: "Gam..." with a value of "0.0" and "Coef." with a value of "0.0". The main area on the right contains a text log of training results, which is repeated four times. The log for each iteration shows: "nSV = 100, nBSV = 100", "optimization finished, #iter = 50", "nu = 1.0", "obj = -100.0, rho = 0.0", "nSV = 100, nBSV = 100", "Total nSV = 150", "Trained patterns: 150", and "Accuracy: 33.33333333333336%". At the bottom left, it says "Task completed." and at the bottom right, there is a "Cancel" button.

SVM Application

Iris

Clear Prepare Run Step Pause

Kernel function

Linear

Polynomial

Radial Basis Fu...

Sigmoid

Gam... 0.0

Coef. 0.0

```
nSV = 100, nBSV = 100
*
optimization finished, #iter = 50
nu = 1.0
obj = -100.0, rho = 0.0
nSV = 100, nBSV = 100
Total nSV = 150
Trained patterns: 150
Accuracy: 33.33333333333336%

*
optimization finished, #iter = 50
nu = 1.0
obj = -100.0, rho = 0.0
nSVTraining...
= 100, nBSV = 100
*
optimization finished, #iter = 50
nu = 1.0
obj = -100.0, rho = 0.0
nSV = 100, nBSV = 100
*
optimization finished, #iter = 50
nu = 1.0
obj = -100.0, rho = 0.0
nSV = 100, nBSV = 100
Total nSV = 150
Trained patterns: 150
Accuracy: 33.33333333333336%
```

Task completed.

Cancel