

Aprendizaje Automático

Departamento de Informática – UC3M

TUTORIAL 3 – Búsqueda hiperparámetros en SVM

2023

1

Tutorial 3

División de las muestras en la práctica (disponibles)



```
X_train, X_test = np.array_split(X, [3650])
y_train, y_test = np.array_split(y, [3650]) #4380-730

# The indices which have the value -1 will be kept in train.
train_indices = np.full((2920,), -1, dtype=int) # 8 years

# The indices which have zero or positive values, will be kept in validation
val_indices = np.full((730,), 0, dtype=int) # 2 years
test_fold = np.append(train_indices, val_indices)
```

2

Tutorial 3

División de las muestras en la práctica (disponibles)



```
from sklearn.model_selection import PredefinedSplit
ps = PredefinedSplit(test_fold)
...
budget = 20
regr = RandomizedSearchCV(DecisionTreeRegressor(random_state=42),
                          param_grid,
                          scoring='neg_mean_absolute_error',
                          cv=ps,
                          n_jobs=1, verbose=1,
                          n_iter=budget)

np.random.seed(42)
regr.fit(X=X_train, y=y_train)
```

3

Tutorial 3

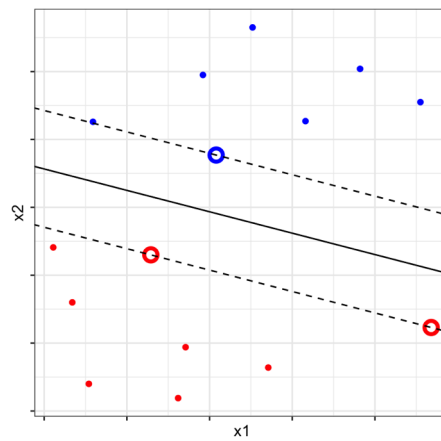
Alternativas Google Colab (ejecución local)

- Opción 3
 - GitHub + tu editor favorito (DataSpell de JetBrains, Visual Studio)
- Opción 4
 - GitHub + Jupyter Notebook (local)
 - Instalación:
 - pip install notebook**
 - Ejecución
 - jupyter notebook**

4

Tutorial 3

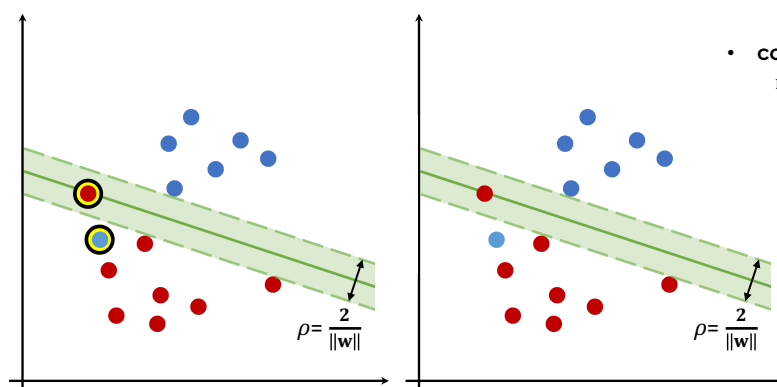
Recordando teoría



5

Tutorial 3

Recordando teoría



Minimizar

- **sin slack**

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

- **con slack**

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum \xi_i$$

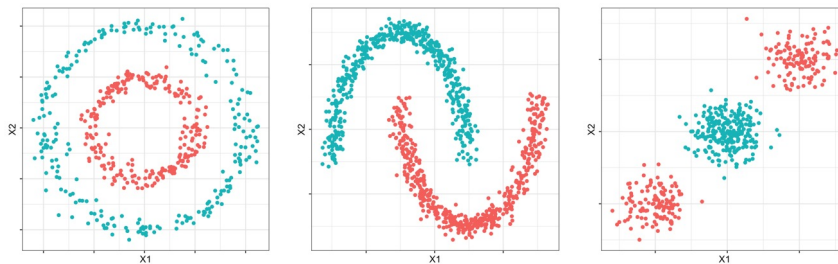
Parámetro C
softSVM

6

Tutorial 3

Recordando teoría

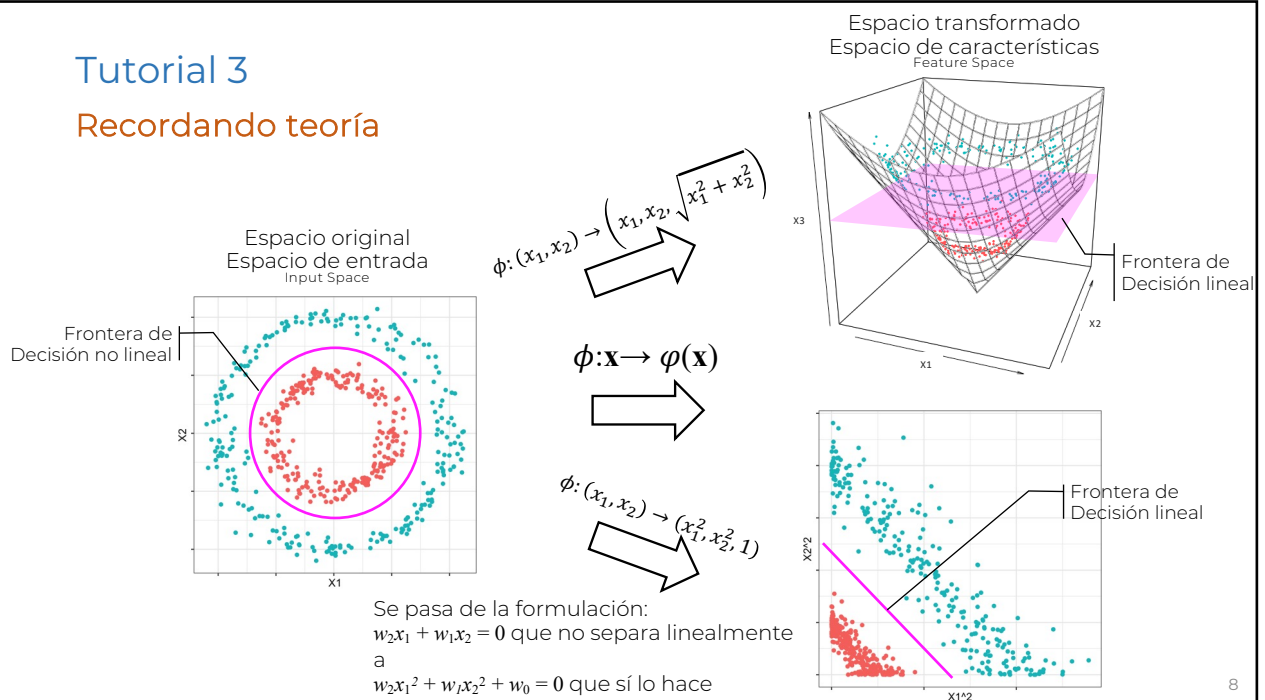
- Modelo lineal
 - los datos son separables linealmente (*hard margin*)
 - los datos no son separables linealmente (*soft margin*)
- Modelo no lineal
 - kernels



7

Tutorial 3

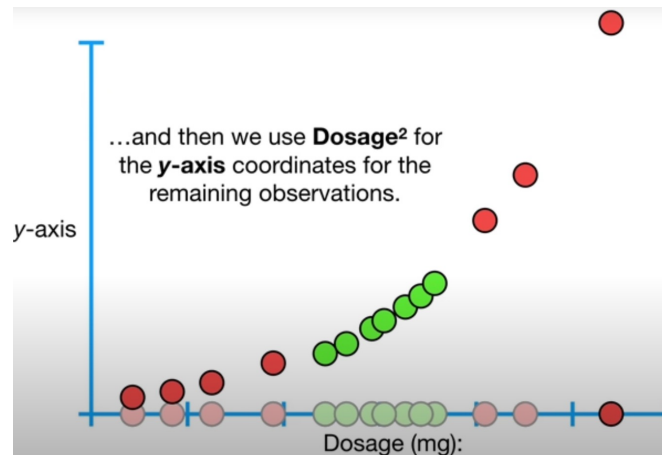
Recordando teoría



8

Tutorial 3

Recordando teoría



9

Tutorial 3

Biblioteca sklearn

- En *Scikit Learn* pueden encontrarse tres implementaciones distintas del algoritmo Support Vector Machine:
 - Las clases `sklearn.svm.SVC` y `sklearn.svm.NuSVC` permiten crear modelos SVM de clasificación empleando kernel **lineal**, **polinomial**, **radial** o **sigmoide**. La diferencia es que **SVC** controla la regularización a través del hiperparámetro **C**, mientras que **NuSVC** lo hace con el número máximo de vectores soporte permitidos.
 - La clase `sklearn.svm.LinearSVC` permite ajustar modelos SVM con kernel lineal. Es similar a **SVC** cuando el parámetro `kernel='linear'`, pero utiliza un algoritmo más rápido.
- Para regresión: `sklearn.svm.SVR`, `sklearn.svm.NuSVR` y `sklearn.svm.LinearSVR`.

10

Tutorial 3

Parámetros

- **C** : float, default=1.0 Parámetro de regularización. La fuerza de la regularización es inversamente proporcional a C. Debe ser estrictamente positivo.
- **gamma** : Se usa en kernel Radial.

$$K(\mathbf{u}, \mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}} = e^{-\gamma\|\mathbf{u}-\mathbf{v}\|^2}$$

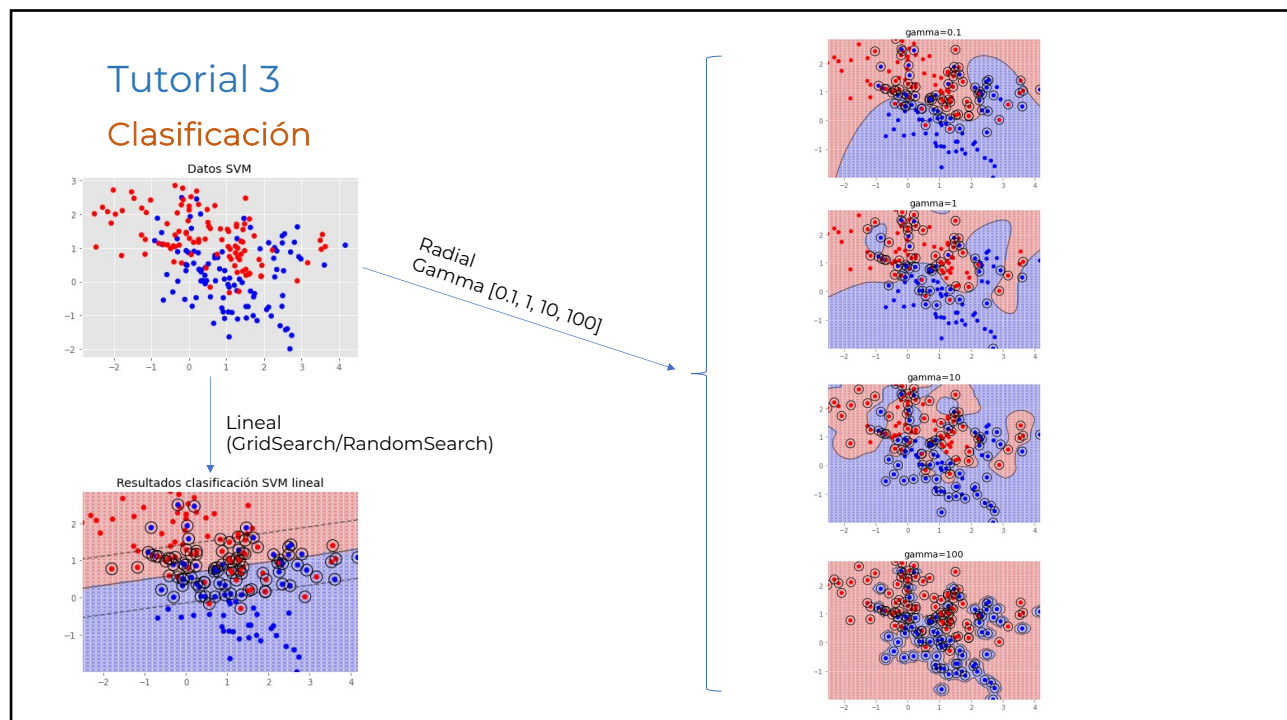
$$\gamma = \frac{1}{2\sigma^2}$$

Define cuánta curvatura queremos en la frontera de decisión:

- Gamma alta significa más curvatura.
- Gamma baja significa menos curvatura.

El parámetro gamma define hasta dónde llega la influencia de un único ejemplo de entrenamiento, donde valores bajos significan 'lejos' y valores altos significan 'cerca'.

11



12

Tutorial 3

Regresión (pipeline)

```
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
# This is the preprocessing pipeline: SVMs need scaling
scaler = StandardScaler()
svr = SVR()
pipe_regr = Pipeline([
    ('scale', scaler),
    ('SVM', svr)])
np.random.seed(42)
pipe_regr.fit(X=X_train, y=y_train)
```

13

Tutorial 3

Regresión (pipeline y GridSearch)

Nombre de parámetros

```
param_grid = {'SVM__C': [0.1, 1, 10, 100],
              'SVM__gamma': [0.01, 0.1, 1]}
inner = KFold(n_splits=3, shuffle=True, random_state=42)
hpo_regr = GridSearchCV(pipe_regr,
                        param_grid,
                        scoring='neg_mean_squared_error',
                        cv=inner, n_jobs=4, verbose=1)
np.random.seed(42)
hpo_regr.fit(X=X_train, y=y_train)
```

14