

Memoria P7.1. Support Vector Machines (SVM)

En esta práctica se ha aplicado el sistema de entrenamiento basado en **Support Vector Machines (SVM)**.

Cómputo de SVM

La función **svm** se utiliza para generar un SVM con los parámetros recibidos, entrenarlo, y, en caso de que se requiera, obtener una posible predicción. Recibe los siguientes parámetros:

- **X_train, y_train**: datos de entrenamiento.
- **X_val, y_val**: datos de validación
- **kernel_type**: tipo de kernel ('linear' o 'rbf' [gaussiano]).
- **c**: factor de control de sobreajuste.
- **sigma**: desviación típica.

```
def svm(X_train, y_train, X_val, y_val, kernel_type, c, sigma):
    # gamma value ignored on linear kernel
    svm = skl_svm.SVC(kernel=kernel_type, C=c, gamma=calc_gamma(sigma))
    svm.fit(X_train, y_train.ravel())

    if X_val is not None and y_val is not None:
        predict = svm.predict(X_val)
        accuracy = skm.accuracy_score(y_val, predict)

    return svm, predict, accuracy
else:
    return svm
```

Obtención parámetros SVM

La función **get_best_svm_parameters** se encarga de seleccionar los mejores parámetros [en este caso, c y sigma] para la SVM.

```
def get_best_svm_parameters(X, y, X_val, y_val):
    values = [0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30]
    accuracies, svm_ = [], []
    for c_value in values:
        for sigma_value in values:
            new_svm, new_accuracy = apply_svm(X, y, X_val, y_val, 'rbf',
            c_value, sigma_value)

            accuracies.append(new_accuracy)
            svm_.append((new_svm, c_value, sigma_value))

    best_accuracy = np.argmax(accuracies)
    svm_, c, sigma = svm_[best_accuracy]

    print("C:", c)
    print("Sigma:", sigma)
    print('Accuracy:', accuracies[best_accuracy])

    ut.plot_svm('./results/graph_c_' + str(c) + '_sigma_' + str(sigma) +
    '.png', X, y, svm_)
```

Resultado

Tras cargar los datos y hacer el cómputo del SVM, obtenemos los siguientes resultados:

