Exercício 2

O exercício consiste na execução de SVM com *kernel* RBF, usando validação cruzada *k-fold* estratificado e Grid Search para estimar os hiperparâmetros. A resolução está na linguagem Python.

Primeiramente, carregam-se os dados numa lista de listas de números reais. As classes dos dados, representados na última coluna, são carregados numa lista de inteiros separada. Os dados estão no arquivo data1.csv. A leitura é feita pelo pacote `csv`. A primeira linha do arquivo é ignorada na leitura por ser de cabeçalho.

```
import csv
dados = []
classes = []
with open('data1.csv', 'r') as arq:
    leitor = csv.reader(arq)
    next(leitor)
    for linha in leitor:
        classes += [int(linha[-1])]
        del linha[-1]
        dados += [[float(i) for i in linha]]
```

Depois, define-se a função de cálculo de acurácia e importam-se os pacotes necessários.

```
import numpy
from sklearn.cross_validation import StratifiedKFold
from sklearn.svm import SVC

def taxa_acertos(v1, v2):
    res = 0.0
    for i in range(0, len(v1)):
        if v1[i] == v2[i]:
            res += 1.0
    return res / len(v1)
```

Então, os dados foram divididos usando 5-*fold*, e cada *fold* foi dividido usando 3-*fold*, que por sua vez foram usados no método Grid Search para o cálculo dos hiperparâmetros. Foi executado SVM com RBF como *kernel* nos *folds* com valores pré-determinados para C e γ, guardando a acurácia de cada execução. Ao final do Grid Search, os valores de C e γ que resultaram na maior acurácia foram usados na execução do SVM nos conjuntos gerados pelo 5-*fold*. Ao final, é apresentada a acurácia média obtida.

Os valores de hiperparâmetros testados foram $C \in \{2^{-5}, 2^{-2}, 2^{0}, 2^{2}, 2^{5}\}$ e $\gamma \in \{2^{-15}, 2^{-10}, 2^{-5}, 2^{0}, 2^{5}\}$.

```
sfinal = StratifiedKFold(classes, n_folds = 5)
for train_index, test_index in sfinal:
```

```
    treino = list(dados[i] for i in train_index)
    cl_treino = list(classes[i] for i in train_index)
    teste = list(dados[i] for i in test_index)
    cl_teste = list(classes[i] for i in test_index)
    coefs, accs = [], []
    sinn = StratifiedKFold(cl_treino, n_folds = 3)
    for train_index, test_index in sinn:
        treinoin = list(treino[i] for i in train_index)
        cl_treinoin = list(cl_treino[i] for i in train_index)
        testein = list(treino[i] for i in test_index)
        cl_testein = list(cl_treino[i] for i in test_index)
        for c in [2**-5, 2**-2, 2**0, 2**2, 2**5]:
            for gamma in [2**-15, 2**-10, 2**-5, 2**0, 2**5]:
                svm_part = SVC(C = c, kernel = 'rbf', gamma = gamma)
                svm_part.fit(treinoin, cl_treinoin)
                res_part = svm_part.predict(testein)
                acc = taxa_acertos(res_part, cl_testein)
                coefs += [[c, gamma, acc]]
                print('C=', c, ' gamma=', gamma, ' acc=', acc)
    coefs.sort(key = lambda x : x[2])
    c, gamma, acc = coefs[-1][0], coefs[-1][1], coefs[-1][2]
    print('Final: C=', c, 'gamma=', gamma, 'acc=', acuracia)
    svm = SVC(C = c, kernel = 'rbf', gamma = gamma)
    svm.fit(treino, cl_treino)
    res = svm.predict(teste)
    acuracia = taxa_acertos(res, cl_teste)
    print('Acurácia: ', acuracia)
    accs += [acuracia]
```

A cada iteração do Grid Search são impressos na tela os valores de C, γ e da acurácia obtidos. Por exemplo, nas primeiras cinco iterações a saída é:

```
C= 0.03125  gamma= 3.0517578125e-05  acc= 0.5669291338582677
C= 0.03125  gamma= 0.0009765625  acc= 0.5669291338582677
C= 0.03125  gamma= 0.03125  acc= 0.5669291338582677
C= 0.03125  gamma= 1  acc= 0.5669291338582677
C= 0.03125  gamma= 32  acc= 0.5669291338582677
```

Ao final de cada Grid Search, é executado o SVM nos conjuntos gerados pelo 5-*fold*. Para cada iteração, os parâmetros resultantes foram:

	C	γ	Acurácia máxima $[0, 1]$
1º *fold*	2^5	2^{-10}	0,9361702127659575
2º *fold*	2^5	2^{-5}	0,875
3º *fold*	2^5	2^{-5}	0,8958333333333334
4º *fold*	2^5	2^{-5}	0,9263157894736842
5º *fold*	2^5	2^{-5}	0,9263157894736842

Para cada *fold*, a acurácia obtida foi:

	Acurácia [0, 1]
1° *fold*	0,875
2° *fold*	0,8958333333333334
3° *fold*	0,9263157894736842
4° *fold*	0,9263157894736842
5° *fold*	0,9361702127659575

Por fim, a acurácia média foi de:

```
Acurácia média:  0.91192702500933187
```

A partir dos resultados, pode-se concluir que os hiperparâmetros a serem usados no classificador final (dados originais) é $C = 2^5$ e $\gamma = 2^{-5}$. Pegando os dados como um conjunto de treino, pode-se executar o SVM sobre eles da seguinte forma:

```
svm = SVC(C = 2**5, kernel = 'rbf', gamma = 2**-5)
svm.fit(dados, classes)
```