In [ ]:

```
【ハイパーパラメタ・チューニング】
問3
② GridSearchCVを使って、問2②を解いてみてください。
```

In [33]:

```
# 問2②で求めた際のC,gammaは、それぞれ 10,1 なので、この値を中心とした数点の
# 組合せをグリッドとして、グリッドサーチを実施する。
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

df = pd.read_csv('train.csv')
df = df.drop(['PassengerId','Age','SibSp','Parch','Ticket','Cabin',
              'Name'], axis=1)
df['Embarked'] = df['Embarked'].fillna('S')
df['Embarked'] = df['Embarked'].map({'S': 0, 'C': 1, 'Q': 2})
df['Sex'] = df['Sex'].apply(lambda x: 0 if x=='male' else 1)
df.head()
```

Out[33]:

	Survived	Pclass	Sex	Fare	Embarked
0	0	3	0	7.2500	0
1	1	1	1	71.2833	1
2	1	3	1	7.9250	0
3	1	1	1	53.1000	0
4	0	3	0	8.0500	0

In [34]:

```
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

params = {
    'C': [0.01, 0.1, 1, 10],
    'gamma': [0.01, 0.1, 1, 10],
    'kernel': ['rbf']
}
grid = GridSearchCV(estimator=SVC(), param_grid=params,
                    cv=5, n_jobs=-1)
grid.fit(x_train, y_train)
```

Out[34]:

```
GridSearchCV(cv=5, estimator=SVC(), n_jobs=-1,
             param_grid={'C': [0.01, 0.1, 1, 10], 'gamma': [0.01,
0.1, 1, 10],
                         'kernel': ['rbf']})
```

In [35]:

```
print('results :{}' .format(grid.cv_results_))
print()
print('best score : {:.5f}' .format(grid.best_score_))
print()
print('best parameters : {}' .format(grid.best_params_))
```

```
results :{'mean_fit_time': array([0.01550431, 0.011836  , 0.013877
01, 0.00995755, 0.01018667,
       0.00859785, 0.00821557, 0.00878663, 0.00906181, 0.00817361,
       0.00955219, 0.00994048, 0.01202083, 0.01189651, 0.01350765,
       0.01039586]), 'std_fit_time': array([0.00326631, 0.00095954
, 0.001075  , 0.00157592, 0.00079506,
       0.00087722, 0.00067515, 0.00056051, 0.00093541, 0.0007466 ,
       0.00153722, 0.0011471 , 0.00179366, 0.00180591, 0.00203988,
       0.0020012 ]), 'mean_score_time': array([0.00506988, 0.00418
835, 0.00458813, 0.00440292, 0.00346355,
       0.00361834, 0.00354805, 0.00336809, 0.00358596, 0.00311055,
       0.00342064, 0.0038393 , 0.00356331, 0.00349226, 0.0031354 ,
       0.00286055]), 'std_score_time': array([0.00118831, 0.000614
82, 0.00084509, 0.00033328, 0.00049878,
       0.00059432, 0.00055363, 0.00034074, 0.00052666, 0.00045966,
       0.00053351, 0.00116754, 0.00024844, 0.00027756, 0.00066782,
       0.0006221 ]), 'param_C': masked_array(data=[0.01, 0.01, 0.0
1, 0.01, 0.1, 0.1, 0.1, 0.1, 1, 1, 1, 1,
                   10, 10, 10, 10],
             mask=[False, False, False, False, False, False, False
, False,
                   False, False, False, False, False, False, False
, False],
       fill_value='?',
            dtype=object), 'param_gamma': masked_array(data=[0.01,
0.1, 1, 10, 0.01, 0.1, 1, 10, 0.01, 0.1, 1, 10,
                   0.01, 0.1, 1, 10],
             mask=[False, False, False, False, False, False, False
, False,
                   False, False, False, False, False, False, False
, False],
       fill_value='?',
            dtype=object), 'param_kernel': masked_array(data=['rbf
', 'rbf', 'rbf', 'rbf', 'rbf', 'rbf', 'rbf', 'rbf',
                   'rbf', 'rbf', 'rbf', 'rbf', 'rbf', 'rbf', 'rbf'
, 'rbf'],
             mask=[False, False, False, False, False, False, False
, False,
                   False, False, False, False, False, False, False
, False],
       fill_value='?',
            dtype=object), 'params': [{'C': 0.01, 'gamma': 0.01, '
kernel': 'rbf'}, {'C': 0.01, 'gamma': 0.1, 'kernel': 'rbf'}, {'C':
0.01, 'gamma': 1, 'kernel': 'rbf'}, {'C': 0.01, 'gamma': 10, 'kern
el': 'rbf'}, {'C': 0.1, 'gamma': 0.01, 'kernel': 'rbf'}, {'C': 0.1
, 'gamma': 0.1, 'kernel': 'rbf'}, {'C': 0.1, 'gamma': 1, 'kernel':
'rbf'}, {'C': 0.1, 'gamma': 10, 'kernel': 'rbf'}, {'C': 1, 'gamma'
: 0.01, 'kernel': 'rbf'}, {'C': 1, 'gamma': 0.1, 'kernel': 'rbf'},
{'C': 1, 'gamma': 1, 'kernel': 'rbf'}, {'C': 1, 'gamma': 10, 'kern
el': 'rbf'}, {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}, {'C': 10,
'gamma': 0.1, 'kernel': 'rbf'}, {'C': 10, 'gamma': 1, 'kernel': 'r
bf'}, {'C': 10, 'gamma': 10, 'kernel': 'rbf'}], 'split0_test_score
': array([0.608, 0.608, 0.608, 0.608, 0.616, 0.616, 0.608, 0.632,
0.624,
       0.704, 0.744, 0.768, 0.72 , 0.752, 0.784, 0.704]), 'split1_
test_score': array([0.608, 0.608, 0.608, 0.608, 0.68 , 0.608, 0.61
6, 0.616, 0.68 ,
       0.808, 0.808, 0.784, 0.792, 0.832, 0.792, 0.8  ]), 'split2_
```

```
test_score': array([0.616, 0.616, 0.616, 0.616, 0.672, 0.632, 0.6
, 0.6  , 0.688,
       0.728, 0.712, 0.728, 0.712, 0.712, 0.712, 0.736]), 'split3_
test_score': array([0.61290323, 0.61290323, 0.61290323, 0.61290323
, 0.66935484,
       0.61290323, 0.61290323, 0.63709677, 0.67741935, 0.76612903,
       0.77419355, 0.73387097, 0.79032258, 0.78225806, 0.77419355,
       0.74193548]), 'split4_test_score': array([0.61290323, 0.612
90323, 0.61290323, 0.61290323, 0.66935484,
       0.62096774, 0.61290323, 0.61290323, 0.67741935, 0.73387097,
       0.79032258, 0.75      , 0.83064516, 0.77419355, 0.79032258,
       0.75      ]), 'mean_test_score': array([0.61156129, 0.61156
129, 0.61156129, 0.61156129, 0.66134194,
       0.61797419, 0.60996129, 0.6196    , 0.66936774, 0.748     ,
       0.76570323, 0.75277419, 0.76899355, 0.77049032, 0.77050323,
       0.7463871 ]), 'std_test_score': array([0.00311991, 0.003119
91, 0.00311991, 0.00311991, 0.02300444,
       0.00818268, 0.00560089, 0.01343029, 0.02301345, 0.03595634,
       0.03412413, 0.02090773, 0.04568102, 0.03923938, 0.02991096,
       0.03102183]), 'rank_test_score': array([12, 12, 12, 12,  9,
11, 16, 10,  8,  6,  4,  5,  3,  2,  1,  7],
      dtype=int32)}

best score : 0.77050

best parameters : {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
```

In [36]:

```
# 問2②と比較するため、テスト用データで予測してみます。
from sklearn.metrics import accuracy_score

clf = SVC(C=10, gamma=1)
clf.fit(x_train, y_train)
pred = clf.predict(x_test)
acc = accuracy_score(pred, y_test)
print('accuracy score : {:.5f}' .format(acc))
```

```
accuracy score : 0.78731
```