Technical Details About Model Saving & Loading

1. Implementation in Python

In python, the pipelined model is defined by a `sklearn.pipeline.Pipeline` instance. The saving & loading of the model is achieved by the `pickle` library, which supports the serialization and de-serialization of runtime object.

Below is the code segment taken from the pipeline code template of the SPACS project, to demonstrate the technical details.

1.1 Model Definition and Training

In [ ]:
```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LassoCV, ElasticNet
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.decomposition import PCA
from sklearn.svm import LinearSVC
from sklearn.datasets import load_iris
from sklearn.feature_selection import SelectFromModel
from sklearn.model_selection import train_test_split

tuned_parameters = [{'kernel': ['rbf'], 'gamma': [10, 1, 1e-1, 1e-2],'C': [0.01, 0.1, 1, 10, 100, 1000]},
                    {'kernel': ['linear'], 'C': [0.01, 0.1, 1, 10, 100, 1000,10000,100000]}]

pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('lasso', SelectFromModel(LassoCV(cv=5), threshold=1e-4)),
    ('pca', PCA(n_components=2)),
    ('grid_search', GridSearchCV(SVC(), tuned_parameters, cv=5)) ])

X_train, X_test, y_train, y_test = train_test_split(X, y)
pipeline.fit(X_train, y_train)
print('Test accuracy: %.3f' % pipeline.score(X_test, y_test))
```

The above sample code defines a pipelined model (a sklearn.pipeline.Pipeline instance), which includes a standard scaler (feature rescaling), a LASSO feature selection module, a PCA dimensionality reduction module and a SVC (support vector classifier) optimized by grid search.

Pipeline.fit() trains the model with provided data.

1.2 Model Saving

In [ ]:
```
import pickle
from sklearn.externals import joblib
joblib.dump(pipeline, 'FILEPATH_PLACEHOLDER.pkl')
```

Persists the trained model (including the definition and trained weights) to a pkl file.
The pickle module implements the binary serializing and de-serializing of a Python runtime object.

1.3 Model Loading

In [ ]:
```
import pickle
from sklearn.externals import joblib
pipeline_r = joblib.load('FILEPATH_PLACEHOLDER.pkl')
pipeline_r.predict(X_test), y_test
```

Restore or load the saved model from the pkl file. After loading, the pipelined model can be used immediately.

2. Implementation in other languages

The workflow profile of SPACS isn't meant to use a specific language or technical implementation. For most modern programming languages, runtime object serialization & deserialization is already supported and the pipelined model saving & loading can be easily implemented.