MYONGJI UNIVERSITY

Artificial Intelligence

Lecture 11. Logistic Regression -- 예제

Spring 2022

Prof. Jonghoon Chun, Ph.D.

E-mail : jchun@mju.ac.kr
Lecture Note: http://lms.mju.ac.kr

Agenda

- Classification concept review
- Linear Regression
- Logistic Regression

LOGISTIC REGRESSION 예제

Logistic Regression -- sklearn

```
from sklearn.linear_model import LogisticRegression
....
Clf = LogisticRegression( )
```

Logistic Regression -- sklearn

- Regularization is applied by default
  - There are different types of regularization methods (e.g., L1, L2 regularization)
  - L1 regularization: Lasso Regression $+\lambda\sum_{j=1}^{n}|\theta_j|$
  - L2 regularization: Ridge Regression $+\lambda\sum_{j=1}^{n}\theta_j^2$

- Parameters
  - Penalty: l1, l2 and others (default = 'l2')
  - Solver(Algorithm to use in the optimization problem)
    - Sag: Stochastic Average Gradient descent
    - Saga: Variant of Sga
    - Liblinear(default): library for Large Linear Classification
      - ✓ Use CD(Coordinate Descent) algorithm
  - max_iter: maximum iteration to converge (default = 100)

Back to movie review…

- Logistic Regression을 사용한 classification

```
from sklearn import metrics
from sklearn.linear_model import LogisticRegression

vect = CountVectorizer().fit(text_train)
X_train = vect.transform(text_train)
X_test = vect.transform(text_test)

clf = LogisticRegression()
clf.fit(X_train, y_train)
pre = clf.predict(X_test)

ac_score = metrics.accuracy_score(y_test, pre)
print("정답률 =", ac_score)
```

```
정답률 = 0.86664
```

Logistic Regression with Cross Validation

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics, model_selection

# logistic regression을 이용
clf = LogisticRegression()

#학습 데이터만을 사용하여 cross valiation
scores = model_selection.cross_val_score(clf, X_train, y_train, cv = 5)

print("Accuracy = ", scores)
print("Average Accuracy = ", scores.mean())
```

```
Accuracy =  [0.8824 0.876  0.8822 0.8882 0.8776]
Average Accuracy =  0.88128
```

Dimension Reduction

- Dimension reduction(차원 축소)
  - Minimum df (document frequency) 제한: 자주 출현하지 않는 토큰을 분석에서 제외하여 feature를 줄임
  - CountVectorizer parameter
    - min_df: minimum document frequency
    - max_df: maximum document frequency
    - Others…

```
vect = CountVectorizer(min_df = 5).fit(text_train)
X_train = vect.transform(text_train)
print("min_df 제한 X_train: {}".format(repr(X_train)))
```

```
min_df 제한 X_train: <25000x27271 sparse matrix of type '<class 'numpy.int64'>'
        with 3354014 stored elements in Compressed Sparse Row format>
```

Dimension Reduction: 74,849 -> 27,271로 줄어듬

Dimension Reduction

```
from sklearn.datasets import load_files
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import metrics, model_selection
from sklearn.linear_model import LogisticRegression

# 데이터 loading
reviews_train = load_files("/Users/jonghoonchun/Downloads/aclImdb/train")
reviews_test = load_files("/Users/jonghoonchun/Downloads/aclImdb/test")

text_train, y_train = reviews_train.data, reviews_train.target
text_test, y_test = reviews_test.data, reviews_test.target

# 불필요한 tag 삭제
text_train = [doc.replace(b"<br />", b" ") for doc in text_train]
text_test = [doc.replace(b"<br />", b" ") for doc in text_test]

# BOW로 학습 및 테스트 데이터 생성
vect = CountVectorizer(min_df = 5, max_df = 1000).fit(text_train)
X_train = vect.transform(text_train)
X_test = vect.transform(text_test)

# logistic regression 활용
clf = LogisticRegression(solver = 'sag', max_iter = 10000)
clf.fit(X_train, y_train)
pre = clf.predict(X_test)

#학습 데이터만을 사용하여 cross valiation
scores = model_selection.cross_val_score(clf, X_train, y_train, cv = 5)

print("Accuracy = ", scores)
print("Average Accuracy = ","%.2f"%scores.mean())
```

```
Accuracy =  [0.843  0.8386 0.849  0.846  0.8382]
Average Accuracy =  0.84
```

```
ac_score = metrics.accuracy_score(y_test, pre)
print("Accuracy = ", ac_score)
```

```
Accuracy =  0.81152
```

MYONGJI UNIVERSITY

END

MYONGJI UNIVERSITY