라쏘(Lasso) 회귀

학습 내용

01. 라쏘(Lasso) 회귀 배경와 원리에 대해 알아본다.

02. 라쏘(Lasso) 회귀 파라미터에 대해 알아본다.

03. 라쏘(Lasso) 회귀와 릿지 회귀의 차이에 대해 알아본다.

(1) 라쏘회귀(Lasso)

라쏘(Lasso) 회귀도 계수를 0에 가깝게 만들려고 한다.

L1 규제라고도 한다.

라쏘는 실제로 계수가 0이 된다. => 모델에서 제외되는 feature(특성)이 생긴다.

In [15]:
```
from sklearn.datasets import load_boston
from sklearn.preprocessing import MinMaxScaler, PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [16]:
```
def load_extended_boston():
    boston = load_boston()  # 데이터 셋 불러오기
    X = boston.data         # 입력 데이터

    X = MinMaxScaler().fit_transform(boston.data)  # 입력 데이터 정규화
    X = PolynomialFeatures(degree=2, include_bias=False).fit_transform(X)
    return X, boston.target
```

In [17]:
```
X, y = load_extended_boston()
print(X.shape, y.shape)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
lr = LinearRegression().fit(X_train, y_train)

print("훈련 데이터 세트 점수 : {:.2f}".format(lr.score(X_train, y_train)))
print("테스트 데이터 세트 점수 : {:.2f}".format(lr.score(X_test, y_test)))
```

```
(506, 104) (506,)
훈련 데이터 세트 점수 : 0.94
테스트 데이터 세트 점수 : 0.79
```

In [29]:
```
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge   # 릿지회귀
import numpy as np

lasso = Lasso().fit(X_train, y_train)
print("훈련 데이터 세트 점수 : {:.2f}".format(lasso.score(X_train, y_train)))
print("테스트 데이터 세트 점수 : {:.2f}".format(lasso.score(X_test, y_test)))
print("사용한 특성의 수 : {:.2f}".format(np.sum(lasso.coef_ != 0)))
```

```
훈련 데이터 세트 점수 : 0.27
테스트 데이터 세트 점수 : 0.26
사용한 특성의 수 : 3.00
```

Lasso는 훈련 세트와 테스트 데이터 세트에서 결과가 좋지 않음.

과소적합이다. 104개의 특성(feature)중에서 4개만 사용하였다.

앞에서는 alpha를 1.0을 사용함.

(2) 라쏘회귀의 alpha의 값

- alpha=0.01로 사용함.
- 이를 위해 max_iter값을 늘려야 한다. 그렇지 않으면 늘리라는 경고가 발생.
- alpha값을 낮추면 모델의 복잡도는 증가한다. 훈련세트와 테스트 세트에서 성능이 향상됨.

- alpha의 값을 너무 낮추면 규제의 효과가 없어져 과대적합이 된다. lm모델과 비슷

In [30]:
```
lasso001 = Lasso(alpha=0.01, max_iter=100000).fit(X_train, y_train)
print("훈련 데이터 세트 점수 : {:.2f}".format(lasso001.score(X_train, y_train)))
print("테스트 데이터 세트 점수 : {:.2f}".format(lasso001.score(X_test, y_test)))
print("사용한 특성의 수 : {:.2f}".format(np.sum(lasso001.coef_ != 0)))
```

```
훈련 데이터 세트 점수 : 0.89
테스트 데이터 세트 점수 : 0.80
사용한 특성의 수 : 34.00
```

In [38]:
```
import matplotlib.pyplot as plt

# 한글
import matplotlib
from matplotlib import font_manager, rc
font_loc = "C:/Windows/Fonts/malgunbd.ttf"
font_name = font_manager.FontProperties(fname=font_loc).get_name()
matplotlib.rc('font', family=font_name)

%matplotlib inline
```

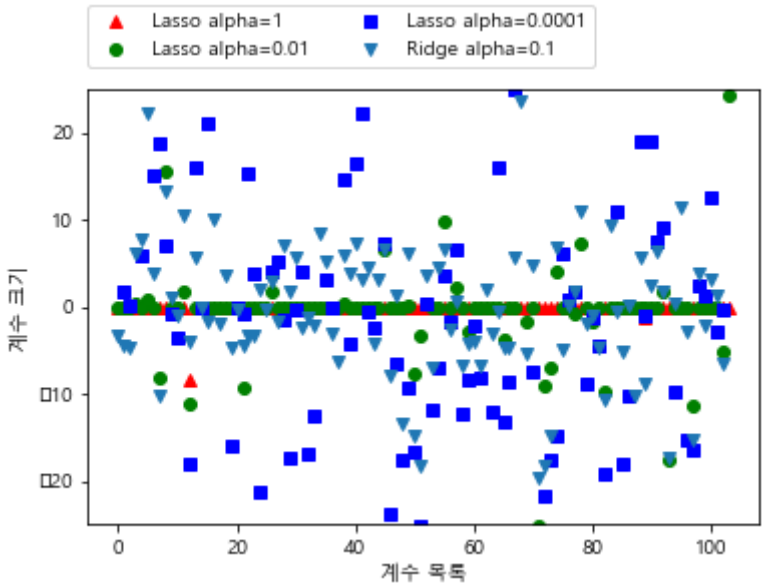
In [39]:
```
lasso = Lasso(alpha=1).fit(X_train, y_train)
lasso001 = Lasso(alpha=0.01).fit(X_train, y_train)
lasso00001 = Lasso(alpha=0.0001).fit(X_train, y_train)
ridge01 = Ridge(alpha=0.1).fit(X_train, y_train)

plt.plot(lasso.coef_, "r^", label="Lasso alpha=1")
plt.plot(lasso001.coef_, 'go', label="Lasso alpha=0.01")
plt.plot(lasso00001.coef_, "bs", label="Lasso alpha=0.0001")

plt.plot(ridge01.coef_, "v", label="Ridge alpha=0.1")

plt.xlabel("계수 목록")
plt.ylabel("계수 크기")
plt.ylim(-25, 25)
plt.legend(ncol=2, loc=(0,1.05))
plt.show()
```

```
C:₩Users₩WITHJS₩Anaconda3₩lib₩site-packages₩sklearn₩linear_model₩coordinate_descent.py:491: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Fitting data with very small alpha may cause precision problems.
  ConvergenceWarning)
```



Lasso alpha =1 일때 거의 0이고 값도 큰 값이 없다.

Lasso alpha =0.01 일때 값이 0이 많고 조금 큰 값도 있음

Lasso alpha =0.0001 계수 대부분이 0이 아니고 값도 커져 꽤 규제받지 않는 모델이 되었음.

- 성능은 Lasso alpha=0.01과 Ridge의 alpha=0.01과 성능이 비슷하다.
- Ridge는 어떤 계수도 0이 되지 않는다.
- 실제로는 릿지 회귀를 선호한다.
- 단, 특성이 많고 그중 일부분만 중요하다면 Lasso가 더 좋은 선택일 수도 있다.

In [ ]: