Forward Selection with `statsmodels`

Thursday April 23, 2015

Python's statsmodels doesn't have a built-in method for choosing a linear model by forward selection. Luckily, it isn't impossible to write yourself. So Trevor and I sat down and hacked out the following. It tries to optimize adjusted R-squared by adding features that help the most one at a time until the score goes down or you run out of features.

```
import statsmodels.formula.api as smf

def forward_selected(data, response):
    """Linear model designed by forward selection.

    Parameters:
    -----------
    data : pandas DataFrame with all possible predictors and response

    response: string, name of response column in data

    Returns:
    --------
    model: an "optimal" fitted statsmodels linear model
           with an intercept
           selected by forward selection
           evaluated by adjusted R-squared
    """
    remaining = set(data.columns)
    remaining.remove(response)
    selected = []
    current_score, best_new_score = 0.0, 0.0
    while remaining and current_score == best_new_score:
        scores_with_candidates = []
        for candidate in remaining:
            formula = "{} ~ {} + 1".format(response,
                                           ' + '.join(selected + [candidate]))
            score = smf.ols(formula, data).fit().rsquared_adj
            scores_with_candidates.append((score, candidate))
        scores_with_candidates.sort()
        best_new_score, best_candidate = scores_with_candidates.pop()
        if current_score < best_new_score:
            remaining.remove(best_candidate)
            selected.append(best_candidate)
            current_score = best_new_score
    formula = "{} ~ {} + 1".format(response,
                                   ' + '.join(selected))
    model = smf.ols(formula, data).fit()
    return model
```

There isn't just one way to design this kind of thing. You could select on some other evaluation metric. You could use internal cross-validation. You might not want to do use forward selection at all. But hey!

Here's how ours can be applied to a classic data set on Discrimination in Salaries:

```
import pandas as pd

url = "http://data.princeton.edu/wws509/datasets/salary.dat"
data = pd.read_csv(url, sep='\\s+')

model = forward_selected(data, 'sl')

print model.model.formula
# sl ~ rk + yr + 1

print model.rsquared_adj
# 0.835190760538
```

- Plan ← Space
- Edit this page
- Find Aaron on
    - Twitter
    - LinkedIn
    - Google+
    - GitHub
    - email
- Comment below
