

Forward Selection και Σύγχρονες Μέθοδοι Επιλογής Μεταβλητών

Στέλιος Αρβανίτης

Τμήμα Οικονομικής Επιστήμης
Ο.Π.Α.

Το οικονομετρικό πρόβλημα I

Θεωρούμε γραμμικό υπόδειγμα:

$$Y_{(i)} = X_{(i)}\beta_0 + \varepsilon_{(i)}, \quad i = 1, \dots, n,$$

όπου:

$$X_{(i)} = (X_{(i)1}, \dots, X_{(i)p}),$$

με μεγάλο αριθμό υποψήφιων παλινδρομητών $p \gg n$.

Στη σύγχρονη οικονομετρία εμφανίζονται συχνά περιβάλλοντα:

- μεγάλος αριθμός παλινδρομητών,
- αβεβαιότητα ως προς το «σωστό» υπόδειγμα,
- πρόβλημα overfitting,
- στόχοι προβλεπτικής ικανότητας αντί ανάκτησης αιτιωδών σχέσεων.

Το οικονομετρικό πρόβλημα II

Το κεντρικό πρόβλημα είναι:

ποιο υποσύνολο από παλινδρομητές να χρησιμοποιηθεί;

Το πρόβλημα αυτό είναι συνδυαστικό:

$$2^p - 1$$

πιθανά υποδείγματα.

Για μεγάλα p , η εξανλητική επιλογή γίνεται υπολογιστικά αδύνατη.

Forward Selection ως greedy αλγόριθμος I

Ο Forward Selection (FS) είναι greedy αλγόριθμος επιλογής μεταβλητών.

Greedy ιδέα:

- ξεκινάμε από κενό υπόδειγμα,
- σε κάθε βήμα προσθέτουμε τη μεταβλητή που βελτιώνει περισσότερο το criterion,
- η επιλογή είναι τοπικά βέλτιστη.

Ο FS είναι εξαιρετικά διαδομένος:

- στην επιστήμη υπολογιστών,
- στο machine learning,
- στη συνδυαστική βελτιστοποίηση.

Παρά τη χρησιμότητά του, η χρήση του στη θεωρητική οικονομετρία παραμένει σχετικά περιορισμένη.

Forward Selection αλγόριθμος

- 1 Αρχικοποίηση:

$$S_0 = \emptyset.$$

- 2 Για κάθε μη επιλεγμένη μεταβλητή:

$$j \notin S_k,$$

υπολογίζουμε τη βελτίωση του criterion.

- 3 Επιλέγουμε:

$$j^* = \arg \max_j \text{improvement}(j).$$

- 4 Σταματάμε όταν:

$$\text{improvement} < \tau.$$

Python implementation: Forward Selection I

```
selected = []
remaining = list(range(p))

while True:

    best_var = None
    best_mse = current_mse

    for var in remaining:

        trial_vars = selected + [var]

        X_trial = sm.add_constant(
            X[:, trial_vars]
        )

        model = sm.OLS(y, X_trial).fit()

        mse = mean_squared_error(
            y,
            model.fittedvalues
        )

        if mse < best_mse:
            best_mse = mse
            best_var = var
```

Python implementation: Forward Selection II

```
improvement = current_mse - best_mse

if improvement < threshold:
    break

selected.append(best_var)
remaining.remove(best_var)

current_mse = best_mse
```

Optimization και submodularity I

Το πρόβλημα της βέλτιστης επιλογής υποσυνόλου είναι γενικά NP-hard.

Ο FS δεν εγγυάται την εύρεση ολικού βέλτιστου.

Παρόλα αυτά, greedy αλγόριθμοι διαθέτουν σημαντικές θεωρητικές ιδιότητες.

Σε αρκετά προβλήματα βελτιστοποίησης εμφανίζεται η έννοια της submodularity.

Μια συνολοσυνάρτηση function $F(S)$ λέγεται submodular όταν εμφανίζει:

”φθίνουσες οικονομίες κλίμακας”.

Δηλαδή:

$$F(S \cup \{j\}) - F(S)$$

μειώνεται καθώς το S μεγαλώνει.

Σε submodular προβλήματα, οι greedy αλγόριθμοι συχνά αποδίδουν λύσεις κοντά στο βέλτιστο.

Η forward selection δεν είναι γενικά ακριβώς submodular στο γραμμικό μοντέλο, αλλά σε αρκετά περιβάλλοντα εμφανίζει περίπου submodular συμπεριφορά-εξαρτάται από την συμπεριφορά των ιδιοτιμών της $X_n'X_n$.

Αυτό εξηγεί γιατί στην πράξη:

- συχνά λειτουργεί αρκετά καλά,
- παραμένει υπολογιστικά πολύ φθηνός,
- και κλιμακώνεται εύκολα σε μεγάλα προβλήματα.

Θεωρήσαμε sparse γραμμικό υπόδειγμα:

$$Y = X\beta_0 + \varepsilon.$$

Χρησιμοποιήθηκε:

$$p = 50$$

υποψήφιοι regressors αλλά μόνο:

$$s = 5$$

πραγματικά σχετικοί regressors.

Η αληθής παράμετρος ήταν:

$$\beta_0 = (2, -1.5, 1.2, 1, -0.8, 0, \dots, 0)'.$$

Οι regressors δημιουργήθηκαν ως:

$$X_{(i)j} \sim N(0, 1),$$

και:

$$\varepsilon_{(i)} \sim N(0, 1).$$

Παράμετροι Monte Carlo:

$$R = 500, \quad n = 120.$$

Stopping rule:

$$\text{improvement} < 0.02.$$

Κριτήριο επιλογής: η αντικειμενική συνάρτηση των εμπειρικών τετραγώνων

Σε κάθε μονοπάτι MC καταγράφηκαν:

- αριθμός επιλεγμένων regressors,
- ποσοστό ανάκτησης των πραγματικών παλινδρομητών (Recovery Rate),
- MC MSE,
- exact recovery indicator.

Ορίσαμε:

$$\text{Recovery Rate} = \frac{\#\{\text{σωστοί selected regressors}\}}{s}.$$

και:

$$\text{Exact Recovery} = \mathbf{1}\{\hat{S} = S_0\}.$$

Κεντρικό ερώτημα:

προβλεπτική ικανότητα vs ανάκτηση δομικού υποδείγματος.

Η διάκριση αυτή είναι θεμελιώδης στη σύγχρονη οικονομετρία και στην μηχανική μάθηση.

Μετρική	Τιμή	Σχόλιο
Average selected size	≈ 10	σημαντικό over-selection
Average recovery rate	1.00	όλοι οι relevant regressors ανακτώνται
Exact recovery rate	χαμηλό	προστίθενται noise regressors
Training MSE	πολύ μικρό	ισχυρό in-sample fit

Κεντρική παρατήρηση

Ο FS:

- βρίσκει πολύ αποτελεσματικά τους ισχυρούς regressors,
- αλλά συνεχίζει να προσθέτει παλινδρομητές "θορύβου",
- επειδή το κριτήριο συνεχίζει να βελτιώνεται ελαφρά.

Άρα:

καλή προβλεπτική ικανότητα $\nRightarrow$ σωστή ανάκτηση της πραγματικής δομής του υποδείγματος.

Ο backward selection λειτουργεί αντίστροφα:

- ξεκινά από το πλήρες μοντέλο,
- αφαιρεί regressors σταδιακά,
- σε κάθε βήμα αφαιρεί τη «λιγότερο χρήσιμη» μεταβλητή.

Συχνά:

- είναι πιο σταθερός,
- αλλά υπολογιστικά ακριβότερος όταν p είναι μεγάλο.
- συνδυάσιμος με τον FS.

Ένα κεντρικό σύγχρονο οικονομετρικό ερώτημα είναι:

πρόβλεψη ή αιτιακή δομή;

Η machine learning προσέγγιση δίνει κυρίως έμφαση:

προβλεπτική ικανότητα.

Η κλασική οικονομετρία ενδιαφέρεται περισσότερο για:

δομική ερμηνεία και ταυτοποίηση.

Συμβιβάσιμα;