UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA jawara

Supervised Learning

#3 Meeting

Shrinkage Method

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 199005202024061001

Kekuatan & Kelemahan OLS

Ordinary Least Squares (OLS) adalah fondasi kita.

Kekuatan:

1. Tujuannya sangat jelas: menemukan koefisien (β) yang meminimalkan RSS (Residual Sum of Squares).
2. Menghasilkan estimator yang Tidak Bias jika asumsi terpenuhi (BLUE - Best Linear Unbiased Estimator).
3. Interpretasinya mudah.

Kelemahan / Asumsi Implisit:

1. Ia akan melakukan apa pun untuk menurunkan RSS, bahkan jika itu berarti menghasilkan model yang tidak masuk akal dan tidak stabil.
2. OLS hanya peduli pada satu hal: akurasi pada data training.

Multikolinieritas

Apa itu Multikolinieritas?

- Kondisi di mana dua atau lebih variabel prediktor (X) dalam model regresi saling berkorelasi kuat.
- Artinya, variabel-variabel tersebut memberikan informasi yang tumpang tindih (redundant).

Analogi: Mencoba memprediksi berat badan seseorang menggunakan prediktor:

- x_1: tinggi badan (cm)
- x_2: tinggi badan (inci)

Kedua variabel ini pada dasarnya mengukur hal yang sama.

Multikolinieritas

Ketika OLS dihadapkan pada multikolinieritas, ia menjadi "bingung" dalam menentukan kontribusi unik dari setiap variabel.

- Estimasi Koefisien (β) Menjadi Sangat Tidak Stabil:
  - Nilai koefisien bisa "meledak" menjadi sangat besar, baik positif maupun negatif.
  - Sedikit saja perubahan pada data (misal, menambah atau menghapus satu baris) bisa menyebabkan nilai koefisien berubah secara drastis.
  - Tanda koefisien (+ atau -) bisa berlawanan dengan intuisi.
- Standar Error Menjadi Sangat Besar:
  - Ini membuat uji hipotesis (p-value) tidak dapat diandalkan. Kita bisa salah menyimpulkan bahwa sebuah variabel tidak signifikan padahal sebenarnya signifikan.

Multikolinieritas

Mengingat Kembali Solusi OLS

Seperti yang telah kita turunkan, solusi untuk koefisien OLS dalam bentuk matriks adalah:

$\hat{\beta}$_ols = $(X^T X)^{-1} X^T y$

Seluruh kestabilan dan keberadaan solusi ini bergantung pada satu komponen krusial: $(X^T X)^{-1}$, yaitu invers dari matriks $X^T X$. Jika komponen ini bermasalah, seluruh solusi OLS akan bermasalah.

Apa Itu Matriks $X^T X$?

Matriks $X^T X$ adalah inti dari perhitungan OLS. Ia berukuran ((k+1) x (k+1)) dan berisi informasi tentang hubungan antar variabel prediktor.

- Elemen diagonalnya berisi jumlah kuadrat dari setiap variabel prediktor.
- Elemen di luar diagonalnya berisi jumlah dari hasil perkalian antar variabel prediktor.

Poin Kunci: Ketika variabel prediktor saling berkorelasi tinggi (multikolinieritas), elemen di luar diagonal dari matriks $X^T X$ ini akan menjadi sangat besar.

Multikolinieritas

Mengapa Multikolinieritas Menghancurkan Invers $(X^TX)^{-1}$?

Dalam aljabar linear, sebuah matriks tidak dapat di-invers jika ia singular. Sebuah matriks menjadi singular jika determinan-nya adalah nol.

1. Hubungan ke Multikolinieritas: Multikolinieritas yang sempurna (misal, $x_1 = 2 * x_2$) menyebabkan kolom-kolom di matriks X menjadi dependen secara linear. Ini secara matematis akan membuat determinan dari (X^TX) menjadi persis nol.
   - Akibat: Karena formula invers matriks melibatkan pembagian dengan determinan (1 / det(A)), kita akan menghadapi pembagian dengan nol. Invers tidak ada, dan OLS gagal total secara matematis.
2. Kasus Dunia Nyata (Multikolinieritas Tinggi, Tidak Sempurna): Di dunia nyata, kita lebih sering menghadapi korelasi yang sangat tinggi, tapi tidak sempurna.
   - Akibat: Determinan dari (X^TX) tidak persis nol, tetapi menjadi sangat sangat kecil (mendekati nol).
   - Ketika kita menghitung invers $(X^TX)^{-1}$, kita membagi dengan angka yang sangat kecil. Hasilnya adalah angka-angka di dalam matriks $(X^TX)^{-1}$ akan "meledak" menjadi sangat besar.

Multikolinieritas

Dampak Akhir pada Koefisien OLS

Karena $\hat{\beta}$_ols = $(X^T X)^{-1} X^T y$, jika matriks $(X^T X)^{-1}$ berisi angka-angka yang sangat besar, maka hasil koefisien $\hat{\beta}$ kita juga akan menjadi sangat besar dan tidak stabil. Sedikit saja perubahan pada data y akan menyebabkan perubahan besar pada $\hat{\beta}$. Inilah penjelasan matematis mengapa koefisien OLS menjadi tidak dapat diandalkan saat ada multikolinieritas.

Overfitting

Pikirkan OLS sebagai seorang siswa yang sangat rajin tetapi tidak berusaha mengerti konsep. Tujuannya hanya satu: mendapatkan nilai 100 di buku latihan.

Analogi Siswa "Hafalan":

- Data Training: Buku latihan berisi 100 soal dan kunci jawabannya.
- Model OLS yang Overfitting: Siswa tersebut tidak belajar rumus atau konsepnya, ia hanya menghafal mati ke-100 soal dan jawabannya. Hasilnya? Jika diuji menggunakan soal dari buku latihan itu, nilainya akan 100 (RSS sangat rendah).
- Data Baru (Ujian Sebenarnya): Ujian berisi 10 soal baru yang tipenya mirip, tetapi angkanya berbeda. Karena siswa ini tidak pernah belajar konsepnya dan hanya menghafal, ia akan gagal total (performa prediksi sangat buruk).

Overfitting

Apa itu Overfitting?

- Terjadi ketika model kita terlalu "menghafal" data training, termasuk noise dan kebetulan acak di dalamnya.
- Hasilnya, model memiliki RSS yang sangat rendah pada data training, tetapi performa yang sangat buruk pada data baru.
- Masalah ini sering terjadi ketika jumlah prediktor (k) sangat banyak dibandingkan jumlah observasi (n).

Overfitting

Kapan OLS Cenderung "Menghafal" (Overfitting)?

1. Ketika Prediktor Terlalu Banyak: Ini penyebab utamanya. Jika kita memiliki 30 titik data ($n=30$) tetapi mencoba menggunakan 25 variabel prediktor ($k=25$), Kita memberikan model terlalu banyak "fleksibilitas". Dengan 25 "alat", model dapat menemukan kombinasi yang sangat rumit untuk mencocokkan setiap titik data dengan sempurna, termasuk noise atau kebetulan acak di dalamnya.
2. Ketika Model Terlalu Kompleks: Bahkan dengan sedikit prediktor, jika Kita memasukkan banyak suku polinomial (x^2, x^3, x^4, ...) atau interaksi, Kita juga memberikan fleksibilitas berlebihan pada model untuk "menghafal".

Overfitting

Bagaimana Overfitting Menyebabkan Varians Tinggi?

Ketika kita menambahkan banyak sekali prediktor ke dalam model (terutama jika beberapa di antaranya berkorelasi), kita menciptakan multikolinieritas buatan.

1. Dengan semakin banyaknya kolom di matriks X, kemungkinan besar beberapa kolom akan menjadi kombinasi linear (atau hampir menjadi kombinasi linear) dari kolom-kolom lainnya.
2. Seperti yang kita buktikan sebelumnya, kondisi ini menyebabkan matriks (X^TX) menjadi hampir singular.
3. Matriks yang hampir singular memiliki determinan yang sangat dekat dengan nol.
4. Ketika kita menghitung inversnya, $(X^TX)^{-1}$, kita membagi dengan determinan yang sangat kecil ini, menyebabkan angka-angka di dalam matriks invers "meledak" menjadi sangat besar.
5. Melihat kembali ke formula varians, $Var(\hat{\beta}) = \sigma^2(X^TX)^{-1}$, jika $(X^TX)^{-1}$ berisi angka-angka raksasa, maka $Var(\hat{\beta})$ juga akan menjadi raksasa.

Regularisasi

Dilema Inti: Akurasi vs. Kompleksitas

OLS memiliki satu tujuan tunggal: memaksimalkan akurasi (meminimalkan RSS).

- Untuk mencapai ini, OLS rela membuat model menjadi sangat kompleks (dengan koefisien yang besar dan tidak stabil), terutama saat menghadapi multikolinieritas atau banyak prediktor.
- Dilema: Apakah kita mau model yang 100% akurat pada data lama tapi tidak berguna untuk prediksi, atau model yang 95% akurat tapi stabil dan bisa diandalkan untuk masa depan?

Regularisasi lahir untuk menyelesaikan dilema ini.

Regularisasi

Filosofi Regularisasi adalah mengubah tujuan dari pemodelan.

Tujuan OLS:

- Temukan koefisien (β) yang MEMINIMALKAN RSS.

Tujuan Regularisasi:

- Temukan koefisien (β) yang MEMINIMALKAN RSS, DENGAN SYARAT koefisien tersebut tidak boleh terlalu besar.

Regularisasi secara sengaja "mengorbankan" sedikit akurasi pada data training untuk mendapatkan model yang jauh lebih sederhana, stabil, dan dapat digeneralisasi pada data baru.

Regularisasi

Bagaimana cara menerapkan "batasan" tersebut secara matematis?
Kita modifikasi fungsi biaya OLS.

$$\text{Fungsi Biaya Baru} = \underbrace{\text{RSS}}_{\text{Ukuran Kecocokan Model}} + \underbrace{\text{PENALTI}}_{\text{Ukuran Kompleksitas Model}}$$

Model sekarang harus melakukan trade-off:

- Jika ia membuat koefisien (β) terlalu besar untuk menurunkan RSS, nilai Penalti akan meningkat.
- Jika ia membuat semua koefisien kecil untuk menurunkan Penalti, nilai RSS akan meningkat.

Tujuannya adalah menemukan keseimbangan terbaik antara kecocokan (fit) dan kesederhanaan (simplicity).

Regularisasi

Suku Penalti adalah sebuah fungsi yang mengukur "besarnya" total dari semua koefisien dalam model.

$$\text{Penalti} = \lambda \times (\text{Fungsi dari besaran semua } \beta_j)$$

- Semakin besar nilai absolut dari koefisien-koefisien (βj), semakin besar nilai Penalti.
- λ (lambda) adalah "kenop" yang kita atur untuk menentukan seberapa kuat hukuman yang akan kita berikan.

Model yang memiliki koefisien besar akan mendapat "hukuman" (nilai penalti yang tinggi), sehingga algoritma akan berusaha mencari solusi dengan koefisien yang lebih kecil.

Shringkage

Karena adanya "hukuman" ini, efek langsung pada koefisien adalah penyusutan (shrinkage).

- Estimasi koefisien dari metode regularisasi secara sistematis akan lebih kecil (lebih dekat ke nol) dibandingkan dengan estimasi koefisien dari OLS.
- Proses "menarik" atau "menyusutkan" koefisien menuju nol inilah yang disebut Shrinkage.
- Manfaatnya adalah mengurangi varians dari estimasi koefisien, sehingga model menjadi lebih stabil.

Shringkage

Pertanyaannya sekarang adalah: Bagaimana cara kita menghitung "besaran total" dari koefisien untuk suku penalti?

Ada dua cara utama yang paling populer, yang masing-masing melahirkan metode regularisasi yang berbeda:

1. Menggunakan Jumlah Kuadrat Koefisien (Penalti L2)
2. Menggunakan Jumlah Nilai Absolut Koefisien (Penalti L1)

Ridge Regression

- Fungsi Penalti: Menggunakan jumlah kuadrat dari koefisien.
- Penalti = $\lambda * \sum(\beta_j)^2$
- Cara Kerja: Hukuman meningkat secara eksponensial dengan besarnya koefisien. Koefisien yang besar akan dihukum jauh lebih berat daripada koefisien yang kecil.
- Efek Shrinkage: Ridge akan menyusutkan semua koefisien mendekati nol, tetapi tidak akan pernah membuatnya menjadi persis nol. Semua variabel tetap ada di dalam model.

LASSO Regression

- Fungsi Penalti: Menggunakan jumlah nilai absolut dari koefisien.
- Penalti = $\lambda * \sum|\beta_j|$
- Cara Kerja: Hukuman meningkat secara linear dengan besarnya koefisien.
- Efek Shrinkage: Lasso juga menyusutkan koefisien. Namun, keistimewaannya adalah ia bisa menyusutkan koefisien hingga menjadi persis nol. Ini berarti Lasso secara efektif dapat melakukan seleksi fitur otomatis, membuang variabel yang dianggap tidak penting.

Summary

- OLS bisa gagal saat menghadapi multikolinieritas dan overfitting karena tujuannya hanya satu: meminimalkan RSS.
- Regularisasi memperkenalkan pendekatan baru dengan menyeimbangkan akurasi (RSS) dan kompleksitas (Penalti).
- Mekanisme ini "menghukum" koefisien yang besar, yang menghasilkan efek Shrinkage (penyusutan koefisien menuju nol).
- Ridge (Penalti L2) menyusutkan semua koefisien secara proporsional.
- Lasso (Penalti L1) menyusutkan koefisien dan bisa menghilangkan variabel yang tidak penting.
- Hasil akhirnya adalah model yang mungkin sedikit bias, tetapi memiliki varians yang jauh lebih rendah, sehingga lebih stabil dan andal untuk prediksi.

SEE YOU NEXT WEEK !

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 199005202024061001
ferdian.bangkit@untirta.ac.id