

Algoritma Genetika

23 Oktober 2020

RRI Pro2 Bengkulu

Daftar isi

- Konsep-konsep dasar Optimisasi
- Algoritma genetika

Konsep Dasar dalam Optimisasi

Optimisasi Matematika

Masalah optimisasi (matematik)

$$\begin{array}{ll} \text{minimize} & f_o(x) \\ \text{subject to} & f_i \leq b_i, i = 1 \dots m \end{array}$$

- $x = [x_1, x_2, \dots, x_n]$: variabel optimisasi
- $f_o: \mathcal{R}^n \rightarrow \mathcal{R}$: Fungsi tujuan
- $f_i: \mathcal{R}^n \rightarrow \mathcal{R}, i = 1, \dots, m$: fungsi kendala

Solusi optimal x^* memiliki nilai terkecil f_o diantara seluruh vector yang memenuhi semua kendala.

Contoh-contoh

Optimisasi portopolio

- Peubah: jumlah yang diinvestasikan ke dalam beberapa aset
- Kendala: total anggaran, investasi minimum/maksimum per aset, untung minimum
- Tujuan: risiko keseluruhan atau variasi untung

Ukuran alat dalam sirkuit elektronika

- Peubah: lebar dan panjang alat
- Kendala: batas manufaktur, kebutuhan waktu, luas maksimum
- Tujuan: konsumsi daya

Penyesuaian data (data fitting)

- Peubah: parameter-parameter model
- Kendala: informasi yang telah diketahui, limit-limit parameter
- Tujuan: kesalahan prediksi

Menyelesaikan masalah-masalah optimisasi

Masalah optimisasi secara umum

- Sangat susah untuk diselesaikan
- Metode-metode yang ada tidak sempurna, seperti membutuhkan waktu yang lama atau tidak selalu menemukan jawaban

Pengecualian: Kelas-kelas masalah tertentu dapat diselesaikan secara efisien dan handal

- Masalah kuadrat terkecil
- Masalah pemrograman linear
- Masalah optimisasi konveks

Kuadrat terkecil

$$\text{minimize } \|Ax - b\|_2^2$$

Menyelesaikan masalah-masalah kuadrat terkecil

- Solusi analitik: $x^* = (A^T A)^{-1} A^T b$
- Algoritma dan software yang efisien dan handal
- Waktu komputasi berbanding lurus dengan $n^2 k (A \in \mathbb{R}^{k \times n})$
- Teknologi yang mapan

Menggunakan kuadrat terkecil

- Mudah dikenal
- Ada sedikit Teknik baku yang menaikkan fleksibilitas

Pemrograman Linear

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, i = 1, \dots, m \end{array}$$

Menyelesaikan program linear

- Tidak ada rumus analitik untuk jawaban
- Software dan algoritma yang efisien dan handal
- Waktu komputasi sebanding dengan n^2m jika $m \geq n$
- Teknologi yang dewasa

Menggunakan program linear

- Tidak segampang untuk dikenal seperti masalah kuadrat terkecil
- Ada sedikit trik-trik yang dapat digunakan untuk mengubah masalah menjadi program linear

Masalah Optimisasi Konveks

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i \quad i = 1, \dots, m \end{array}$$

- Fungsi tujuan dan kendala bersifat konveks:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

jika $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.

- Meliputi masalah kuadrat terkecil dan program linear sebagai kasus khusus

Menyelesaikan masalah optimisasi konveks

- Tidak ada jawaban analitik
- Algoritma yang efisien dan handal
- Waktu komputasi sebanding dengan $\max(n^3, n^2m, F)$ dimana F adalah biaya untuk menilai f_i dan turunan pertama dan kedua mereka
- Hampir menjadi teknologi

Menggunakan optimisasi konveks

- Sering sulit untuk dikenali
- Banyak trik untuk mengubah masalah menjadi masalah konveks
- Ternyata banyak masalah yang bisa dipecahkan melalui optimisasi konveks

Optimisasi nonkonveks

- Optimisasi konveks memerlukan syarat yang ketat
- Mayoritas masalah bersifat nonkonveks, misalnya yang melibatkan peubah yang diskrit
- Salah satu metode optimisasi untuk masalah nonkonveks adalah algoritma genetika.

Algoritma Genetika

Algoritma Genetik

Algoritma genetik adalah salah satu metode untuk mencari solusi masalah optimisasi nonkonveks. Metode ini meniru proses evolusi pada sistem biologi.

Komponen utama algoritma genetik

- Encoding: penerjemahan variable menjadi gen
- Pembentukan populasi awal
- Evaluasi terhadap individu dalam populasi: fungsi tujuan
- Proses evolusi
 - Cross over
 - Mutation
 - Seleksi : evaluasi + eliminasi

Contoh

- Bagilah angka-angka berikut menjadi tiga kelompok sehingga selisih jumlah angka setiap kelompok dengan kelompok lain nol atau sekecil mungkin! Misalkan angka-angka tersebut adalah 1,2,3,4,5,6.
- Salah satu jawaban dari permasalahan tersebut adalah
 - $R=[1,6], S=[2,5], T=[3,4]$.
 - $J_R = 1 + 6 = 7, J_S = 2 + 5 = 7, J_T = 3 + 4 = 7$
 - Nilai rata-ratanya adalah $J = \frac{(J_R + J_S + J_T)}{3}$
 - Fungsi tujuan= jumlah kuadrat dari simpangan
 - $f_0 = (J_R - J)^2 + (J_S - J)^2 + (J_T - J)^2$
-

Encoding

Encoding adalah proses penyusunan informasi yang efektif dan bisa diolah oleh computer.

- Hasil dari *encoding* adalah gen dari setiap individu.
- Gen digunakan untuk mengevaluasi setiap individu. Hasil evaluasi adalah *fitness value*.
- $R=[1,6], S=[2,5], T=[3,4] \Leftrightarrow RSTTSR$
- $R=[1,2,3], S=[4,5], T=[6] \Leftrightarrow RRRSST$

Pembentukan populasi awal

Populasi awal dibuat dengan menentukan ukuran populasi N_p . Setelah itu, individu dibuat dengan gen yang dipilih secara acak.

- Individu diwakili oleh gennya. Sekumpulan gen kita sebut sebagai kromosom. Misalnya:

RRRRRR, RRRRRS, RRRRRT, RRRRSR, RRRRST

- Dalam contoh kita, jumlah kombinasi gen yang tersedia adalah

$$3 \times 3 \times 3 \times 3 \times 3 \times 3 = 3^6 = 729$$

RRRRRR, RRRRRS, RRRRRT, RRRRSR, RRRRSS, RRRRST, RRRRTR, ..., TTTTTTTT

Evaluasi Individu

Setiap individu dievaluasi untuk menilai kelayakannya.

- Fungsi tujuan digunakan dengan input gen masing-masing individu
- Outputnya adalah nilai kelayakan (*fitness value*)
- $R=[1,6], S=[2,5], T=[3,4] \Leftrightarrow RSTTSR \quad f_0(RSTTSR) = 0$
- $R=[1,2,3], S=[4,5], T=[6] \Leftrightarrow RRRSST \quad f_0(RRRSST) = ???$
 - $J_R = 1 + 2 + 3 = 6, J_S = 4 + 5 = 9, J_T = 6, J = \frac{6+9+6}{7} = 7$
 - $f_0 = (J_R - J)^2 + (J_S - J)^2 + (J_T - J)^2 = (6 - 7)^2 + (9 - 7)^2 + (6 - 7)^2 = 6$
- $R=[1,2,3,4,5,6], S=[], T=[] \Leftrightarrow RRRRRR \quad f_0(RRRRRR) = 0$
 - $J_R = 1 + 2 + 3 + 4 + 5 + 6 = 21, J_S = 0, J_T = 0, J = \frac{21+0+0}{7} = 7$
 - $f_0 = (J_R - J)^2 + (J_S - J)^2 + (J_T - J)^2 = (21 - 7)^2 + (0 - 7)^2 + (0 - 7)^2 = 294$

Cross-Over

- Cross over adalah peristiwa dimana dua individu bertukar gen
- Contoh: RRRRRR dan TTTTTT bertukar tiga gen seperti berikut

R-R-R-T-T-R

T-T-T-R-R-T

- Individu mengalami cross over dipilih secara acak berdasarkan peluang cross over p_c , $0 \leq p_c \leq 1$
- Berapa gen yang tertukar dan lokasi pertukaran bisa dipilih secara acak atau tidak.
- Cross over menghasilkan individu baru. Akibatnya, jumlah individu dalam populasi bertambah.

Mutasi

- Mutasi adalah peristiwa yang menyebabkan gen yang dimiliki oleh individu berubah.
- Individu yang mengalami mutasi dipilih secara acak berdasarkan peluang mutasi p_m , $0 \leq p_m \leq 1$
- Lokasi gen yang berubah juga dipilih secara acak.
- Jumlah gen yang berubah bisa tetap bisa acak.
- Contoh:
 - RRRRRR menjadi RRTRRR: gen pada posisi ketiga berubah dari R menjadi T
- Mutasi menghasilkan individu baru. Akibatnya, jumlah populasi bertambah.

Seleksi

- Ukuran populasi yang diinginkan adalah N_p
- Cross over dan mutasi menambah jumlah individu populasi menjadi kurang lebih

$$N_p + p_c N_p + p_m N_p > N_p$$

- Evaluasi terhadap setiap individu dalam populasi dilakukan.
- Pemilihan N_p dilakukan berdasarkan nilai kelayakan setiap individu.
- Individu yang “jelek” akan dieliminasi. Setelah eliminasi, jumlah individu dalam populasi adalah N_p

Evolusi

- Evolusi adalah peristiwa dimana cross-over, mutasi dan seleksi terjadi berkali-kali.
- Satu putaran yang terdiri dari cross-over, mutasi, dan seleksi disebut satu generasi.
- Jumlah generasi N_g menunjukkan berapa lama evaluasi terjadi. Angka ini ditentukan pada awal proses algoritma genetika.

Pseudocode

- Encoding
- Tentukan N_p, N_g, p_c, p_g
- Inisialisasi populasi : pembuatan dan evaluasi individu baru.
- $i=0$ (jumlah generasi = 0)
- While $i < N_g$:
 - Cross-over
 - Mutasi
 - Seleksi
 - $i = i + 1$
- Solusi = gen individu terbaik

Daftar Pustaka

1. Boyd, S., Vandenberghe, L., *Convex Optimization*, 2004, Cambridge University Press <https://web.stanford.edu/~boyd/cvxbook/>
2. Whitley, D. A genetic algorithm tutorial. *Stat Comput* **4**, 65–85 (1994). <https://doi.org/10.1007/BF00175354>

Kesimpulan

- Konsep dasar dalam optimisasi
- Algoritma genetika

Terima kasih

Novalio Daratha, S.T., M.Sc., Ph. D.
Lektor pada Prodi Teknik Elektro
Universitas Bengkulu



+6281285198719 (phone and whatsapp)

ndaratha@unib.ac.id

<https://github.com/novatha/optimisasi> (Temukan file presentasi ini disana)