

Obliczenia i systemy inspirowane biologicznie

WYKŁAD 3

Inteligencja stadna

- **Inteligencja stadna** (grupowa) (ang. *swarm intelligence*) – podejście do obliczeń oparte o samoorganizującą się populację autonomicznych jednostek oddziałujących na siebie oraz na środowisko, w którym się znajdują.

Algorytmy mrówkowe

- Algorytmy mrówkowe zostały opracowane przez Marco Dorigo.
- W algorytmach mrówkowych autonomicznymi jednostkami są sztuczne mrówki, które szukając rozwiązań w przestrzeni rozwiązań pozostawiają ślad feromonowy.
- Dla obiecujących miejsc w przestrzeni rozwiązań następuje wzrost natężenia feromonu.
- Dla miejsc w przestrzeni rozwiązań o niskiej jakości natężenia feromonu maleją.

Grupowanie danych – podstawowe założenia

- Podstawowym zadaniem grupowania (klasteryzacji) jest dokonanie podziału zbioru przypadków C znajdujących się w bazie na grupy C_1, C_2, \dots, C_k , nazywane klastrami, stanowiące podzbiory przypadków podobnych do siebie, przy czym pojęcie podobieństwa może być definiowane w różny sposób.
- Podział zbioru C powinien być dokonany w taki sposób, aby przypadki z danej grupy były bardziej podobne do siebie (*homogeniczność*) niż do jakichkolwiek przypadków z pozostałych grup (*heterogeniczność*).

Grupowanie danych – podstawowe założenia

- W wielu sytuacjach przypadki grupowane są razem ze względu na istniejące pomiędzy nimi zależności takie jak np. nieodróżnialność, podobieństwo, bliskość, funkcjonalność, zgodność.
- Podobieństwo przypadków może być definiowane na różne sposoby, zależnie od typu grupowanych danych. Bardzo często jako miarę podobieństwa wykorzystuje się odległość pomiędzy przypadkami w przestrzeni, w której rozważane są przypadki.

Miary odległości

- Dane są dwa punkty x oraz y w przestrzeni m -wymiarowej:

$$x = [x_1, x_2, \dots, x_n]$$

$$y = [y_1, y_2, \dots, y_n]$$

Miary odległości

- Odległość (metryka) Euklidesa

$$d_{Eukl}(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

- Odległość (metryka) Manhattanu

$$d_{Manh}(x, y) = \sum_{i=1}^n |y_i - x_i|$$

Miary odległości

- Odległość (metryka) Minkowskiego

$$d_{Mink}(x, y) = \sqrt[q]{\sum_{i=1}^n |y_i - x_i|^q}$$

Miary odległości

- Dla atrybutów symbolicznych możemy zdefiniować funkcję R („różne od”).
- Dla i -tego atrybutu funkcja ma postać:

$$R(x_i, y_i) = \left\{ \begin{array}{ll} 0 & \text{dla } x_i = y_i \\ 1 & \text{w przeciwnym przypadku} \end{array} \right\}$$

- Funkcja R może zostać zastosowana dla i -tego atrybutu w miarach odległości.

Normalizacja wartości atrybutów

- Przy wyznaczaniu odległości atrybuty posiadające duże wartości mogą niwelować wpływ innych atrybutów (tych posiadających mniejsze wartości).
- W celu wyeliminowania tej sytuacji należy dokonać normalizacji wartości atrybutów.

Normalizacja wartości atrybutów

- Normalizacja min-max:
 - dla zbioru wartości atrybutu:

$$X = \{ x_1, x_2, \dots, x_k \}$$

- znormalizowana wartość x_i jest obliczana jako:

$$x_i^* = \frac{x_i - \min(X)}{\text{zakres}(X)} = \frac{x_i - \min(X)}{\max(X) - \min(X)}$$

Algorytmy iteracyjnej optymalizacji

- W algorytmach iteracyjnej optymalizacji najlepszy podział zbioru przypadków jest wyznaczany przez iteracyjne polepszanie pewnych wskaźników jakości, startując z początkowego, najczęściej losowego, podziału.

Mrówkowe algorytmy grupowania

- Zaletą mrówkowych algorytmów grupowania jest brak konieczności określania *a priori* liczby grup, na które zbiór przypadków jest dzielony.
- Mrówkowe algorytmy grupowania są głównie oparte na wersjach zaproponowanych przez:
 - Jean-Louisa Deneubourga,
 - Erika Lumera i Baldo Faieta.

Mrówkowy algorytm grupowania - przykład

- Funkcja sąsiedztwa:

$$f_n(x_i) = \begin{cases} \frac{1}{\sigma^2} \sum_{j \in N} \left(1 - \frac{d(x_i, x_j)}{\alpha} \right) & \text{jeśli } \forall_{j \in N} \left(1 - \frac{d(x_i, x_j)}{\alpha} \right) > 0 \\ 0 & \text{w przeciwnym razie} \end{cases}$$

N – sąsiedztwo i -tego obiektu

σ – rozmiar sąsiedztwa N

α – parametr definiujący skalę
niepodobieństwa obiektów

Mrówkowy algorytm grupowania - przykład

- Prawdopodobieństwo podniesienia obiektu:

$$P_{pick}(x_i) = \left\{ \begin{array}{ll} 1 & \text{jeśli } f_n(x_i) \leq 1 \\ \frac{1}{f_n^2(x_i)} & \text{w przeciwnym razie} \end{array} \right\}$$

- Prawdopodobieństwo upuszczenia obiektu:

$$P_{drop}(x_i) = \left\{ \begin{array}{ll} 1 & \text{jeśli } f_n(x_i) \geq 1 \\ \frac{1}{f_n^4(x_i)} & \text{w przeciwnym razie} \end{array} \right\}$$

Mrówkowy algorytm grupowania - przykład

ALGORYTM

- losowo ustaw obiekty (przypadki) na siatce prostokątnej
- **dla** każdej mrówki a_j :
 - losowo wybierz obiekt x_i
 - ustaw obiekt x_i jako niesiony przez mrówkę a_j
 - ustaw mrówkę a_j na losowo wybranej pustej pozycji na siatce prostokątnej

Mrówkowy algorytm grupowania - przykład

- dla każdej iteracji t od 1 do t_{max} :
 - losowo wybierz mrówkę a_j
 - przesuń mrówkę a_j na losowo wybraną nową pozycję na siatce prostokątnej
 - oblicz prawdopodobieństwo upuszczenia obiektu niesionego przez mrówkę a_j

Mrówkowy algorytm grupowania - przykład

– **jeśli** obiekt zostanie upuszczony przez mrówkę a_j

- **dopóki** mrówka a_j nie niesie żadnego obiektu

- losowo wybierz obiekt x_i

- oblicz prawdopodobieństwo podniesienia obiektu x_i przez mrówkę

a_j

- **jeśli** mrówka podniesie obiekt x_i

- ustaw obiekt x_i jako niesiony przez mrówkę a_j

Mrówkowe algorytmy grupowania

- Podstawowy mrówkowy algorytm grupowania wykorzystuje losowe przemieszczanie się mrówek na siatce prostokątnej.
- W określaniu przemieszczania się mrówek może zostać wykorzystany mechanizm feromonów.

Mrówkowe algorytmy grupowania

- Prawdopodobieństwo przejścia mrówki z komórki i do komórki k :

$$P_{i \rightarrow k} = \frac{W(\varphi_k) w(\Delta_k)}{\sum_{j \in N} W(\varphi_j) w(\Delta_j)}$$

N – sąsiedztwo komórki i

φ – parametr powiązany z natężeniem feromonu

Δ – parametr powiązany z różnicą w orientacji mrówek w kolejnych iteracjach

W, w – funkcje wagowe

Optymalizacja rojem cząstek

- Optymalizacja rojem cząstek (ang. *Particle Swarm Optimization, PSO*) jest metodą poszukiwania najlepszego rozwiązania w N wymiarowej przestrzeni rozwiązań.
- Optymalizacja rojem cząstek została zaproponowana przez Russella Eberharta i Jamesa Kennedy'ego w 1995 roku.

Optymalizacja rojem cząstek

- Najlepsze rozwiązanie poszukiwane jest przez zbiór cząstek lecących jak rój przez przestrzeń rozwiązań.
- Poprzez analogię do stada ptaków podążającego za przywódcą, rój cząstek podąża za najlepszym do tej pory znalezionym rozwiązaniem.

Optymalizacja rojem cząstek

- Każda i -ta cząstka traktowana jest jako punkt w N wymiarowej przestrzeni rozwiązań opisany przez wektor współrzędnych:

$$X^i = (x_1^i, x_2^i, \dots, x_N^i)$$

- Dla każdej i -tej cząstki pamiętana jest jako wektor najlepsza do tej pory znaleziona pozycja (pozycja, dla której funkcja dopasowania miała największą wartość):

$$P^i = (p_1^i, p_2^i, \dots, p_N^i)$$

Optymalizacja rojem cząstek

- Pamiętana jest jako wektor globalnie najlepsza do tej pory znaleziona pozycja:

$$G = (g_1, g_2, \dots, g_N)$$

- Z każdą i -tą cząstką powiązany jest wektor jej prędkości:

$$V^i = (v_1^i, v_2^i, \dots, v_N^i)$$

- Na początku wektory pozycji cząstek oraz wektory ich prędkości inicjalizowane są losowo.

Optymalizacja rojem cząstek

- W każdym kroku algorytmu iteracyjnego modyfikowane są wektory: pozycji cząstek oraz prędkości cząstek zgodnie z zależnościami:

$$v_j^i(t+1) = \omega v_j^i(t) + \psi_1 R_1(p_j^i(t) - x_j^i(t)) + \psi_2 R_2(g_j(t) - x_j^i(t))$$

$$x_j^i(t+1) = x_j^i(t) + v_j^i(t+1)$$

ω, ψ_1, ψ_2 – stałe

R_1, R_2 – dwa wektory wartości losowych z przedziału $[0, 1]$

generowane niezależnie w każdej iteracji dla każdej cząstki $j=1, 2, \dots, N$

Optymalizacja rojem cząstek

- Warunkiem zakończenia algorytmu iteracyjnego może być:
 - osiągnięcie odpowiednio dużej wartości funkcji dopasowania,
 - osiągnięcie maksymalnej liczby iteracji.

Optymalizacja rojem cząstek

- Inteligencja stadna realizowana jest przez pamiętanie globalnie najlepszej do tej pory znalezionej pozycji G .
- Wartości współrzędnych pozycji G uwzględniane są przy wyznaczaniu nowych wartości w wektorach pozycji oraz prędkości cząstek.
- W porównaniu z obliczeniami ewolucyjnymi w optymalizacji rojem cząstek nie występuje proces selekcji. Wszystkie cząstki biorą udział w poszukiwaniu najlepszego rozwiązania.