Informatyka, studia dzienne, mgr II st. semestr I

Obliczenia inteligentne **2010/2011**

Prowadzący: Michał Pryczek Środa, 14:30

Data oddania: _____ Ocena: _____

Leszek Wach 169513
Michał Janiszewski 169485

Zadanie: Analiza danych biomedycznych*

1. Cel

Zadanie polegało na analizie danych biomedycznych uzyskanych dzięki technice płytek genowych (mikromacierzy). Zadanie składało się z dwóch etapów: redukcji ilości cech opisujących osobnika oraz stworzenia klasyfikatora który byłby w stanie określić na podstawie zredukowanego zbioru cech przynależność osobnika do jednej z klas. W naszym przypadku druga część zadania polegała na stworzeniu klasyfikatora potrafiącego rozpoznać, który typ napoju zawierających kwasy tłuszczowe wypiła badana osoba. W doświadczeniu wykorzystano dwa typy napojów: PUFA i SUFA.

2. Wprowadzenie

2.1. Micromacierze

Technika mikromacierzy pozwala na jednoczesną analizę ekspresji tysięcy genów. W celu stworzenia mikromacierzy pobierane są dwie grupy komórek: testowa i odniesienia. Z pobranych komórek zostają wyodrębnione łańcuch mRNA. Ponieważ mRNA łatwo ulega degradacji poddaje się je procesowi odwrotnej transkrypcji, otrzymując komplementarne cDNA.

* SVN: `https://serce.ics.p.lodz.pl/svn/labs/oi/mpat_sr1430/wacjan/OI-MP/@182`

Następnie cDNA zostaje zaznaczone barwnikiem różnym dla komórek testowych i odniesienia. Barwnik pozwala zidentyfikować obecność cząsteczki na płytce. Jako barwnika najczęściej używa się farb fluorescencyjnych.

Kolejnym etapem jest hybrydyzacja polegająca an zanurzeniu płytki w roztworze zabarwionego cDNA. Cząsteczki cDNA łączą się z punktami (sondami) na płytce zawierającymi komplementarne sekwencje DNA.

Po zakończeniu hybrydyzacji płytka zostaje umieszczona w skanerze gdzie następuje odczyt ekspresji genów czyli zabarwienia punktów płytki [1].

2.2. Metoda k-średnich

W zadaniu użyto metody k-średnich. Jest to jedna z metod grupujących pozwalająca znaleźć grupy genów o zbliżonej ekspresji. Metoda polega na utworzeniu pewnej liczby grup i przypisania elementów do grup bazując na pewnej mierze podobieństwa. Obiekty podobne do siebie powinny być umieszczane w tej samej grupie. Metoda dąży do minimalizacji różnic wewnątrz grup i maksymalizacji różnic między różnymi grupami[2]

Opis algorytmu

1. Wyznaczenie k punktów wyznaczających początkowe centroidy grup.
2. Przypisanie każdego obiektu do grupy, dla której odległość między jej centroidem a danym obiektem jest najmniejsza według określonej miary.
3. Wyliczenie nowych centroidów w powstałych grupach np. jako średnią arytmetyczną obiektów w grupie.
4. Powtarzanie kroków 2 i 3 aż do zadanej liczby iteracji lub do momentu, gdy w trakcie iteracji żaden z obiektów nie zmieni grupy.

Metoda k-średnich wymaga znania z góry liczby grup, na które zostanie podzielony zbiór. Wyznaczenie początkowych centroidów może odbyć się losowo i tak też było w naszym programie.

2.3. Miary podobieństwa

W metodzie k-średnich do wyznaczania wartości podobieństwa dwóch wektorów używamy poniższych miar.

Odległość euklidesowa:

$$e(x, y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{1}$$

[1] [1]
[2] [1]

Współczynnik korelacji Pearsona, znany również jako wpsółczynnik korelacji Spearmana:

$$p(x,y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{2}$$

gdzie $\bar{x}$ to wartość średnia wektora x, a $\bar{y}$ wartość średnia wektora y.

2.4. Wielowarstwowy Perceptron

Perceptron jest przykładem jednokierunkowej sieci, w której neurony są ułożone w warstwy. Każda warstwa składa się z pewnej określonej liczby neuronów. Pierwsza warstwa składa się z neuronów kopiujących czyli wyjście każdego neuronu jest takie samo jak jego wejście. Warstwę pierwszą określa się jako warstwę wejściową. Liczba neuronów w warstwie wejściowej określa liczbę wejść sieci. Następnie w sieci znajduje się szereg warstw ukrytych. Ostatnia warstwa jest warstwą wyjściową. Ilość neuronów w warstwie wyjściowej określa liczbę wyjść sieci. Sieć jest tak zorganizowana, że wyjście każdego neuronu w warstwie k jest połączone ze wszystkimi neuronami w warstwie $k+1$ (z wyjątkiem warstwy wyjściowej).

2.5. Nauka sieci

Algorytm wstecznej propagacji błędu jest wykorzystywany w celu minimalizacji błędu sieci. Jest to metoda określana jako nauka z nauczycielem. Nauka z nauczycielem polega na przekazywaniu do sieci pewnych wartości i porównywaniu wyników sieci z pewnymi wartościami oczekiwanymi. W trakcie tego typu nauki obliczany jest błąd wyniki i wagi neuronów sieci są modyfikowany w taki sposób aby w następnej iteracji błąd był mniejszy.

W przypadku sieci wielowarstwowych mamy informacje tylko o błędzie popełnionym przez ostatnią, wyjściową warstwę. W takiej sytuacji stosuje się algorytm wstecznej propagacji błędu. Polega on na tym, że najpierw wyznaczamy wartości błędu dla wszystkich neuronów z ostatniej warstwy o numerze K:

$$\delta_i^K = y_i - r_i \tag{3}$$

gdzie i jest numerem neuronu ($i = 1, 2, ..., N_K - 1$, N_K – liczba neuronów w K-tej warstwie), y_i – wartość oczekiwana na i-tym neuronie, r_i – wartość otrzymana na danym neuronie. Następnie obliczamy wartości błędów dla neuronów w pozostałych warstwach sieci:

$$\delta_i^{K-1} = \sum_{j=1}^{N_K} w_i^{jK}\delta_j^K \tag{4}$$

gdzie w_i^{jK} oznacza i-ty element wektora wag neuronu o numerze j z K-tej warstwy ($i = 1, 2, ..., N_{K-1}$, N_{K-1} jest liczbą neuronó w $K-1$-szej warstwie). Aby zwiększyć zbieżność algorytmu stosuje się współczynnik momentum α:

$$\Delta w_i^{jK(k)} = \eta\delta_i^{K-1}u_i^K + \alpha\Delta w_i^{jK(k-1)} \tag{5}$$

gdzie: $\Delta w_i^{jK(k)}$ – zmiana wagi i w j-tym neuronie w K-tej warstwie w k-tej iteracji, u_i^K – i-ta wartość wejściowa dla K-tej warstwy, η – współczynnik uczenia, α – współczynnik momentum. Obliczoną wartość zmiany wag dodajemy do aktualnych wag poszczególnych neuronów.

3. Opis implementacji

W ramach zadania powstała aplikacja napisana w języku C++ z wykorzystaniem biblioteki Qt. Program wczytuje dane z pliku tekstowego i przechowuje je w obiekcie klasy `MicroArray`. Do prawidłowego wczytania danych wymagane jest ustalenie liczby grup oraz ich liczebności w pliku z danymi.

Po wczytaniu danych zostaje uruchomiona metoda `run()` na rzecz obiektu klasy `Kmeans`. Metoda ta implementuje algorytm k-średnich. Do prawidłowego działania metody wymagane jest ustalenie liczby grup oraz maksymalnej liczby iteracji. Wynikiem metody są liczebności grup oraz lista ostatecznych centroidów należących do tych grup. Na podstawie centroidów program wybiera przedstawicieli poszczególnych grób. Przedstawiciele grup tworzą dane treningowe wykorzystywane w procesie nauki sieci.

Za naukę i testowanie sieci odpowiada klasa `Utils`. W programie wykorzystano bibliotekę FANN 2.1.0 zawierającą implementację perceptronu.

4. Materiały i metody

Badania zostały wykonane na zestawie oznaczonym indeksem GSE13466[3]. Zestaw zawiera profile genów pobranych z komórek krwi obwodowej od 21 młodych mężczyzn.
W zestawie można wyróżnić cztery klasy.

- 6 godzin przed spożyciem PUFA (wielonienasycone kwasy tłuszczowe)
- 6 godzin po spożyciu PUFA
- 6 godzin przed spożyciem SFA (nasycone kwasy tłuszczowe)
- 6 godzin po spożyciu SFA

Metodę k-średnich przetestowano dla wartości k równego: 100, 200 oraz 400. We wszystkich przypadkach liczba iteracji wynosiła 100. Wyniki grupowania można zobaczyć na rysunkach do 1 do 5.

Po przetestowaniu metody k-średnich przeprowadzono 4 testy obejmujące stworzenie klasyfikatora. Algorytm k-średnich był za każdym razem wykonywany oddzielnie dla grupy SUFA i grupy PUFA. W każdym z testów uczono sieci z momentum równym 0.6 i współczynnikiem nauki równym 0.6. Jako algorytm nauki wykorzystano dostarczony z biblioteką FANN algorytm FANN_RPROP. Funkcje aktywacji warstwy ukrytej i wyjściowej ustawiono na FANN_SIGMOID_SYMETRIC.

W trakcie analizy wyników za poprawny wynik klasyfikatora (sieci) uznawano wartości różniące się od wartości oczekiwanej nie więcej niż o 0.01.

[3] [2]

Ogólny przebieg testów:

- uruchomienie algorytmu k-średnich,
- wybranie przedstawicieli wygenerowanych grup,
- zapisanie danych w formacie umożliwiającym naukę sieci, czyli to co wcześniej było kolumnami macierz teraz jest wektorami wejściowymi sieci, określenie oczekiwanych wyjść sieci (1 dla PUFA, -1 dla SUFA),
- podzielenie zbioru danych treningowych na cztery rozłączne zbiory,
- uruchomienie nauki czterech sieci neuronowych z których każda ma inny zbiór walidujący (jeden z czterech wyznaczonych w poprzednim kroku), wszystkie sieci razem stanowią klasyfikator,
- policzenie średniego błędu dla wszystkich sieci, jest to średni błąd stworzonego klasyfikatora,
- przetestowanie klasyfikatora na całym zbiorze danych.

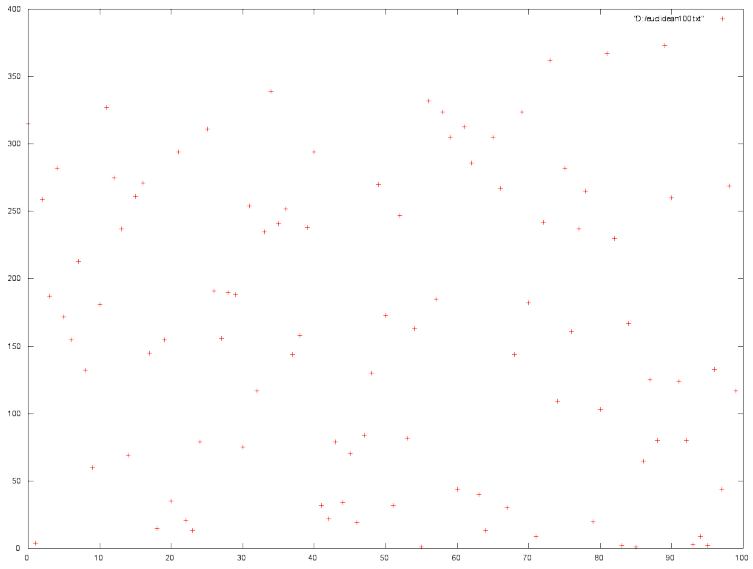
Test1
liczba grup = 50
liczba iteracji algorytmu k-średnich = 100
liczba sieci = 4
liczba neuronów warstwy ukrytej = 10
Wektor wejściowy przekazywany do sieci neuronowej składał się z wartości genów zarówno przed jak i po spożyciu kwasów tłuszczowych.

Test2
liczba grup = 400
liczba iteracji algorytmu k-średnich = 100
liczba sieci = 4
liczba neuronów warstwy ukrytej = 30
Wektor wejściowy przekazywany do sieci neuronowej składał się z wartości genów zarówno przed jak i po spożyciu kwasów tłuszczowych.

Test3
liczba grup = 400
liczba iteracji algorytmu k-średnich = 100
liczba sieci = 4
liczba neuronów warstwy ukrytej = 20
Wektor wejściowy przekazywany do sieci neuronowej składał się tylko z wartości genów po spożyciu kwasów tłuszczowych.

Test4
liczba grup = 600
liczba iteracji algorytmu k-średnich = 100
liczba sieci = 4
liczba neuronów warstwy ukrytej = 35
Wektor wejściowy przekazywany do sieci neuronowej składał się z wartości genów zarówno przed jak i po spożyciu kwasów tłuszczowych.

Wyniki wszystkich testów przedstawiono w tabeli 1.



Rysunek 1. Wykres przedstawiający liczbę cech przyporządkowanych poszczególnym grupom dla $k = 100$, Odległość Euklidesa

Tabela 1. Wyniki testów programu

Numer testu	**Średni błąd klasyfikatora**	**Procent poprawnie sklasyfikowanych próbek**	**Procent poprawnie sklasyfikowanych PUFA**	**Procent poprawnie sklasyfikowanych SUFA**
1	6.011e-05	64.00	100.00	28.57
2	7.660e-15	100.00	100.00	100.00
3	1.607e-14	100.00	100.00	100.00
4	4.936e-14	100.00	100.00	100.00

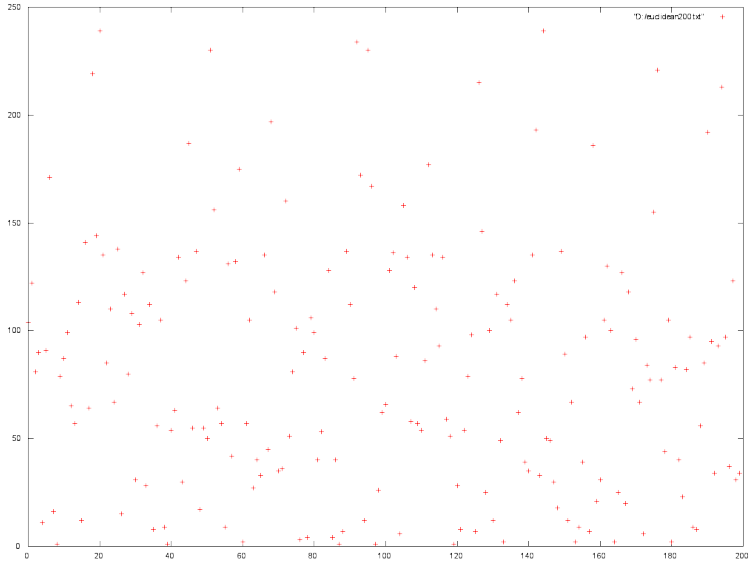
5. Wyniki

Na rysunkach od 1 do 5 przedstawiono wyniki działania algorytmu k-średnich. W tabeli 1 zamieszczono wyniki testów klasyfikatorów uzyskanych w trakcie realizacji zadania.

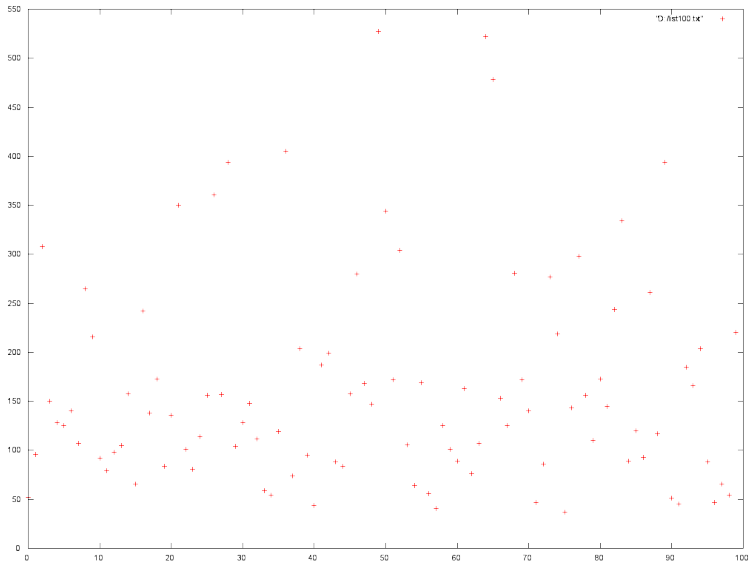
6. Dyskusja

Porównując wyniki działania metody k-średnich można zauważyć, że dla współczynnika Pearsona liczebności poszczególnych grup są bardziej wyrównane niż dla odległości Euclidesa. W przypadku odległości Euclidesa występuje więcej grup z wieloma elementami, jak i grup, które mają zaledwie kilka elementów.

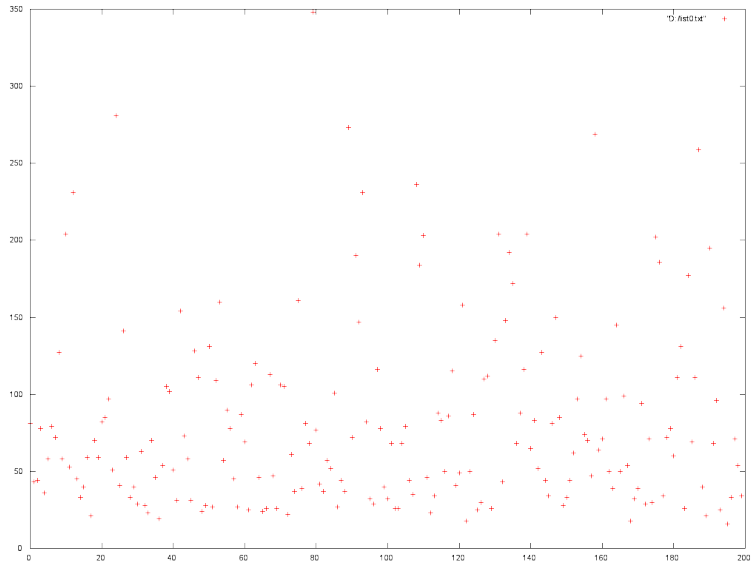
We wszystkich przeprowadzonych testach uzyskano klasyfikatory dające bardzo dobre rezultaty. Większość próbek została sklasyfikowana poprawnie. W przypadku pierwszego testu liczba grup będąca wynikiem algoryt-



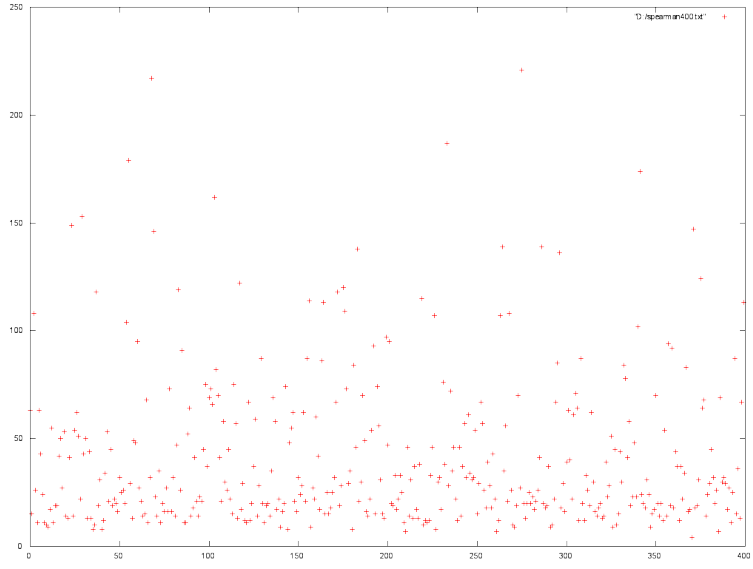
Rysunek 2. Wykres przedstawiający liczbę cech przyporządkowanych poszczególnym grupom dla $k = 200$, Odległość Euklidesa



Rysunek 3. Wykres przedstawiający liczbę cech przyporządkowanych poszczególnym grupom dla $k = 100$, Współczynnik korelacji Pearsona



Rysunek 4. Wykres przedstawiający liczbę cech przyporządkowanych poszczególnym grupom dla $k = 200$, Współczynnik korelacji Pearsona



Rysunek 5. Wykres przedstawiający liczbę cech przyporządkowanych poszczególnym grupom dla $k = 400$, Współczynnik korelacji Pearsona

mu k-średnich okazała się zbyt mała aby poprawnie sklasyfikować wszystkie próbki. W pierwszym teście błąd klasyfikatora jest największy.

W kolejnych trzech testach uzyskaliśmy wyłącznie poprawne rezultaty, przy czym wyniki zwracane przez sieci w teście trzecim miały wartości ok. 0.99 lub -0.99 natomiast w testach 2 i 4 zwracane wartości były równe 1.0 lub -1.0. Oznacza to że minimalnie lepsze rezultaty można uzyskać przy pomocy sieci neuronowych uczonych wektorami składającymi się zarówno z wartości genów przed spożyciem jak i po spożyciu kwasów tłuszczowych.

Literatura

[1] *Metody klasyfikacji we wspomaganiu diagnostyki nowotworów techniką płytek genowych*, [online]. [dostęp: 12 maja 2011]. Dostępny w Internecie: http://microarray.republika.pl/index.html

[2] *Series GSE13466*, [online]. [dostęp: 12 maja 2011]. Dostępny w Internecie: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE13466