H-Adviser

Hybrydowy system wspomagania decyzji inwestycyjnych

Projekt	**H-ADVISER**
Dokument	Cel, założenia oraz obsługa hybrydowego systemu wspomagania decyzji inwestycyjny H-Adviser
Autor	Krzysztof Drelczuk
Data utworzenia	2008-09-22

Wersja	**Data**	**Autor**	**Opis**
0.1	2008-09-22	Krzysztof Drelczuk	Dotyczy wersji alpha 1.0

1. Cel programu.

Program powstał, jako narzędzie do zweryfikowania następującej hipotezy: „Finansowe szeregi czasowe posiadają charakterystyczne wzorce, po których następują wzrosty lub spadki ich wartości.” W celu weryfikacji powyższej hipotezy został stworzony automatyczny system transakcyjny oparty na znalezionych wzorcach. Hipotezę uzna się za prawdziwą, gdy dla danego szeregu czasowego konsekwentne podążanie za sygnałami płynącymi z programu przyniesie stopę zwrotu większą niż strategia buy-and-hold.

2. Podstawowe założenia wyszukiwania wzorców.

Do wyszukiwania wzorców w szeregach czasowych wykorzystano klasteryzację, (czyli bez wzorcową klasyfikację) szeregów czasowych. Klasyfikacja szeregów czasowych przyciąga uwagę coraz większej rzeszy naukowców i praktyków. Szczególnie widoczne jest to przy przetwarzaniu długich szeregów czasowych jakie występują na przykład w bioinformatyce czy też w sferze finansowej. Klasteryzacja jest jedną z najczęściej używanych technik pozyskiwania informacji z dużej ilości danych, o których charakterze ze względu na bardzo dużą wielowymiarowość mamy nie wielkie pojęcie.

Podstawowym celem klasteryzacji jest transformacja danych źródłowych do postaci bardziej kompaktowej, która jest w stanie w pełni je odzwierciedlić. Polega na podziale zbioru danych na odpowiednie klasy abstrakcji, czyli mniejsze podgrupy, gdzie elementy danego klastra są podobne do siebie a mocno odmienne od innych. Rozwój technologii informatycznej w ostatnich latach spowodował duży postęp w wykorzystywaniu narzędzi sztucznej inteligencji w metodach pozyskiwania wiedzy z dużych wielowymiarowych struktur danych.

Problem klasteryzacji jest złożoność obliczeniowa. Jest ona równoważna z optymalizacją globalną nieliniowych funkcji wielomodalnych tak więc zalicza się do problemów NP-trudnych. Pomimo tego metody eksploracji danych finansowych mające na celu predykcje czy też klasyfikacje potrafią osiągać bardzo dużą skuteczność mierzoną tak w wskaźnikach statystycznych jak i realnych korzyściach

ekonomicznych. Teoria błądzenia losowego rynków finansowych, która przekreślałaby skuteczność powyższej analizy szeregów czasowych również była i jest szeroko dyskutowana w wielu pracach. Jednakże w większości przypadku nie dało się jednoznacznie dowieść hipotezy o słabej efektywności rynku według Famy. W szczególności nie udało się tego ustalić dla polskich rynków kapitałowych.

Problem postawiony w celu weryfikacji przez system H-Adviser brzmi: „Czy można tak podzielić szeregi czasowe, aby jeden klaster zawierał te, które w przyszłości wykażą tendencje wzrostowe, a drugi tendencje zniżkowe?"

Jednym z kluczowych problemów podczas procesu klasteryzacji jest wielowymiarowość danych wejściowych. Przestrzeń rozwiązań, którą system będzie dzielić na klastry ma taką wymiarowość jak wektory do niej wprowadzane. W przypadku gdy będą brane dane z sześćdziesięciu (dane z jednej minuty w odstępach jednosekundowych) ostatnich obserwacji, przestrzeń jaką będzie musiał podzielić będzie przestrzenią 60-wymiarową. Klasyfikacja polega na przypisywaniu elementów do odpowiednich klastrów za pomocą z góry założonej normy (na przykład Euklidesowej). Przy dużej liczbie wymiarów różnica pomiędzy najbliższym i najdalszym sąsiadem staje się coraz mniej istotna, co znacznie utrudnia podział przestrzeni na znaczące klastry.

W systemie H-Adviser zostawano falkową redukcję wymiarów. Dyskretna transformacja falkowa jest bardzo częstą techniką wykorzystywaną we wstępnej analizie danych. Dzięki niej można zarówno zredukować ilość wymiarów wektora wejściowego, do docelowego systemu analizującego dane (np. klasyfikującego albo predykcyjnego), jak i usunąć część informacji uznanych przez transformacje, jako szum lub redundancje danych, jednakże pozostawiając w sygnale informacje oryginalne (w sensie Shanon'a).

Do systemu wprowadzane są wektory utworzone na podstawie podłączonego strumienia danych za pomocą okien przesuwnych. Długość okna jest ustalona na 56 obserwacji. Do budowy systemu wspomagania decyzji na drodze eksperymentów został wybrany następujący model falkowej kompresji danych: wektor przekazany do systemu zostanie poddany dyskretnej transformacji wejściowej (DTF) za pomocą falki D2 w sposób następujący:

1. Pierwsze osiem elementów zostanie zredukowanych do dwóch;
2. Kolejne szesnaście elementów zostanie zredukowane do czterech;
3. Trzydzieści dwa następne elementy zostanie zredukowane do ośmiu.

Uzyska się w ten sposób redukcję z przestrzeni 56-wymiarowej do 14-wymiarowej.

3. Aplikacja H-Adviser

3.1 Założenia ogólne

H-Adviser został napisany w języku C# i działa na platformie .NET. Została ona utworzona przez Microsoft i zapewnie niezależność kodu od systemu, na którym będzie on wykonywany. Uruchomienie jego wymaga jednak zainstalowania środowiska wykonawczego, które można pobrać bezpośrednio ze strony `www.micriosoft.com` (Microsoft .NET Framework Version 2.0 Redistributable Package Bezpośredni odnośnik: http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en). W sytuacji, gdy używanym systemem jest MS Vista wtedy nie jest to koniecznie. System ten ma standartowo zainstalowane wyżej wymienione środowisko uruchomieniowe.

Aplikacja jest wielojęzyczna. W tej chwili obsługiwane języki to: polski, francuski i angielski.

3.2. Dane wejściowe

Dane wejściowe powinny być plikiem tekstowych o następującym formacie:

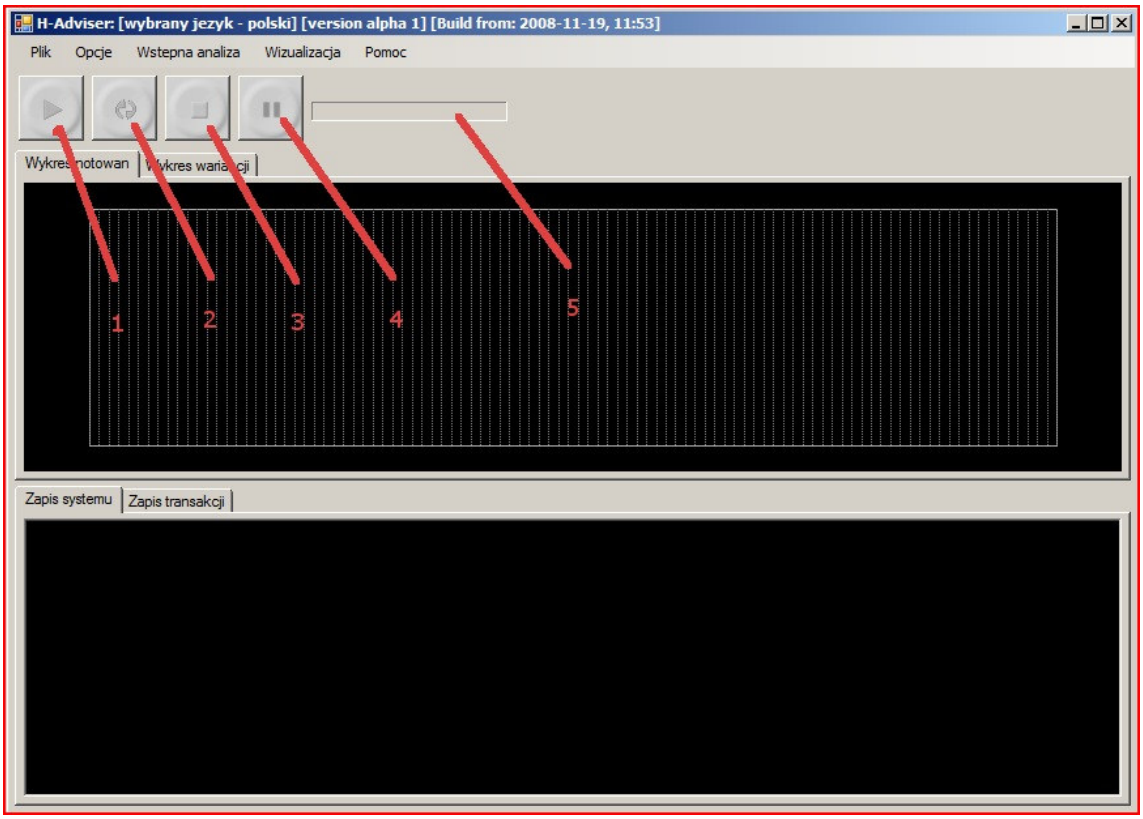
```
<ticker>,<per>,<date>,<time>,<open>,<high>,<low>,<close>,<vol>
WIG20_I,1,20030220,100100,1106.53,1106.53,1106.53,1106.53,0
WIG20_I,1,20030220,100200,1107.23,1107.34,1107.23,1107.34,0
WIG20_I,1,20030220,100300,1107.93,1107.93,1107.93,1107.93,0
WIG20_I,1,20030220,100400,1108.30,1108.30,1108.30,1108.30,0
WIG20_I,1,20030220,100500,1108.05,1108.90,1108.05,1108.90,0
WIG20_I,1,20030220,100600,1109.33,1109.91,1109.33,1109.91,0
WIG20_I,1,20030220,100700,1110.50,1110.50,1110.50,1110.50,0
WIG20_I,1,20030220,100800,1110.50,1110.50,1110.17,1110.17,0
WIG20_I,1,20030220,100900,1110.08,1110.08,1109.40,1109.40,0
WIG20_I,1,20030220,101000,1108.88,1108.88,1108.47,1108.47,0
```

Pierwsza kolumna <ticker> oznacza nazwę waloru, w tym przypadku indeks WIG 20. Kolumna <date> określa datę nadejścia tiku w formacie rrrr-mm-dd, natomiast kolumna <time> okręśla czas w formacie hhmmss. Kolejne cztery kolumny

określają odpowiednio cenę otwarcia, najwyższą i najwyższą w okresie agregacji czasowej oraz cenę zamknięcia. W przypadku minutowych danych tikowych wszystkie te wartości są równe. Ostatnia kolumna oznacza wolumen transakcji.

3.3. Aplikacja H-Adviser

3.3.1. Obsługa podstawowa



1 – Start systemu – aktywne po wczytaniu danych

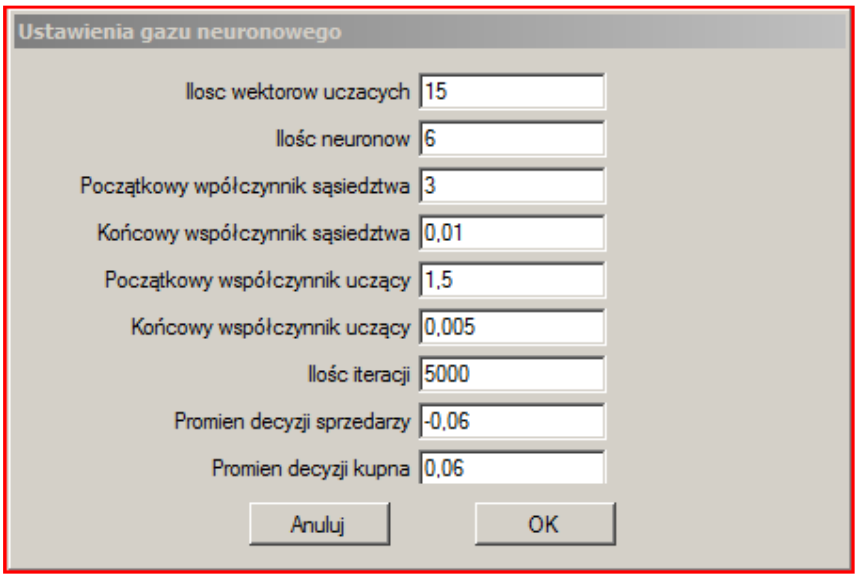
2 – Rozpoczęcie klasyfikacji od początku

3 – Zatrzymanie klasyfikacji

4 – Chwilowe wstrzymanie klasyfikacji

5 – Wskaźnik postępu klasyfikacji dla szeregów czasowych wczytanych z pliku.

3.3.2. Gaz neuronowy



- Ilość wektorów uczących: nie istnieje idealna recepta na dobór tego parametru. Każdy neuron reprezentuje odpowiedni obszar w przestrzeni rozwiązań. Zbyt mała liczba będzie powodowała problem błędnej klasyfikacji wektorów, zbyt duża natomiast stworzy klasy bez reprezentantów, co spowoduje błędy przy uogólnianiu wiedzy.
- Ilość wektorów uczących: ilość wektorów, które będą przekazane do procesu uczenia.
- Współczynnik sąsiedztwa: Funkcja sąsiedztwa przyjmuje postać (3.3.2.a). Jako λ_i oznaczone jest sąsiedztwo początkowe a jako λ_f końcowe. Parametry t oznacza numer wykonywanej iteracji uczenia, natomiast t_{max} jest ustaloną ilością iteracji.

$$\lambda(t) = \lambda_i \left(\frac{\lambda_f}{\lambda_i} \right)^{\frac{t}{t_{\max}}} \qquad (3.3.2.a)$$

Przy $\lambda = 0$ mamy do czynienia z zerowym sąsiedztwem i regułą WTA (winner takes all). Przy $\lambda > 0$ są aktualizowane wagi wszystkich neuronów, (czyli również tych, które są „dalekimi" sąsiadami), lecz współczynnik uczenia odległych sąsiadów szybko dąży do zera. Jest to sytuacja podobna do tej, z jaką

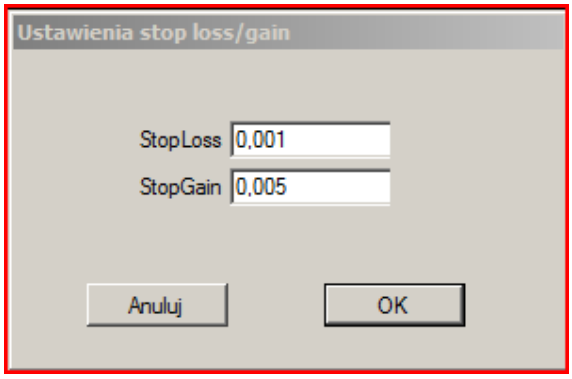
mamy do czynienia w klasycznym algorytmie SOM z gaussowską funkcją sąsiedztwa.

- Współczynnik uczenia: na ogół przyjmuje się, że współczynnik ten maleje ze wzrostem numeru iteracji algorytmu od wartości początkowej ε_i do końcowej ε_f osiąganej w ostatnim przejściu. W systemie przyjmuje się że współczynnik ten maleje zgodnie z funkcją potęgową (3.3.2.b).

$$\varepsilon(t) = \varepsilon_i \left(\frac{\varepsilon_f}{\varepsilon_i} \right)^{\frac{t}{t_{\max}}}, \qquad 3.3.2.b$$

- Ilość iteracji: oznacza ile razy wektory kodowe będą wprowadzone do systemu.
- Promień podjęcia decyzji: jak blisko centrum klastra musi znaleźć się wektor, aby był do niego zaklasyfikowany.

3.3.3. Stop-loss/stop-gain



Wartości stop są ustalane dynamicznie zgodnie z regułą trailing-stop. Edycja ich następuje poprzez podanie wartości procentowej (1=100%).

3.3.4. Komunikaty systemu

Zapis systemu przechowuje informacje o zdarzeniach wygenerowanych przez moduł klasteryzujący. W tej wersji system komunikuje się za pomocą czterech komunikatów:

```
2008-05-30 17:53:14 > Otrzymano wartość: 12,25
Przygotowano wektor do wstępnej analizy:
      12,3  12,3  12,3  12,3  12,25 12,25 12,25 12,25 12,2  12,2  12,2
      12,25 12,2  12,2  12,2  12,2  12,25 12,25 12,25 12,25 12,25 12,25 12,2
```

```
      12,25 12,25 12,25 12,25 12,25 12,25 12,25 12,25 12,3  12,25 12,25
      12,25 12,25 12,25 12,25 12,25 12,25 12,25 12,25 12,25 12,25 12,2
      12,25 12,2  12,25 12,2  12,2  12,2  12,25 12,25 12,3  12,25 12,25
Przygotowano wektor do klasyfikacji:
      0,353 -1    0,795 -0,03 0,353 1     0,353 0,294 0,795 -0,03 0,353
      0,353 0,353
Wektor przyjęto (klasa: 0,39)
```

Pierwsza linijka określa czas, w którym nadeszła wartość (składowana szeregu czasowego) oraz jej wielkość. Po uzyskaniu wystarczającej ilości danych tworzony jest wektor wejściowy i przekazywany jest on do wstępnej analizy danych, co prezentuje linia druga linia (z powodów edycyjnych w pracy została przedstawiona w siedmiu wierszach). Trzecie linijka (tutaj zapisana w trzech wierszach) przedstawia znormalizowany wektor po wstępnej analizie danych. Widać tutaj wyraźnie redukcje wymiarów wektora wejściowego. Ostatnia linia informuje, do jakiej klasy należy wektor oraz czy został on przyjęty czy odrzucony. System przyjmuje, iż dla wartości ujemnych wektor koduje wzorzec sprzedaży a dla wartości dodatnich „kupno" wzorzec kupna.

Zapis transakcji przechowuje informacje o transakcjach hipotetycznie przeprowadzonych przez inwestora, który konsekwentnie podążałby za sygnałami systemu.

Logi można zapisać do pliku tekstowego klikając na nim prawym klawiszem myszy i wybierając opcje „Zapisz do pliku…".