Skalowanie Wielowymiarowe Algorytmy MDS

Krzysztof Bazner
Instytut Matematyczny, Uniwersytet Wrocławski

23/03/2020

1 Wstęp

1.1 Czym jest MDS?

Skalowanie wielowymiarowe (Multidimensional Scaling – MDS) reprezentuje dane dotyczące bliskości (tj. miary podobieństwa, bliskości, powiązania itp.) jako odległości między punktami w przestrzeni wielowymiarowej (zwykle dwuwymiarowej). Skalowanie rozpoczyna się od pewnej konfiguracji początkowej. Punkty są następnie przesuwane iteracyjnie, dzięki czemu dopasowanie między odległościami a danymi jest poprawiane, dopóki dalsze ulepszenia nie wydają się możliwe. Istnieją programy komputerowe do MDS, takie jak Systat lub Proxscal. Im dokładniej dane odpowiadają odległościom w przestrzeni MDS, tym lepiej konfiguracja MDS punktu reprezentuje strukturę bliskości. Jeśli dopasowanie rozwiązania MDS jest dobre, można je sprawdzić wizualnie, próbując je zinterpretować pod względem treści. Popularnym podejściem do tego jest poszukiwanie wymiarów, zazwyczaj głównych osi, które mają sens pod względem tego, co jest znane lub zakładane na temat obiektów reprezentowanych przez punkty.

1.2 Co omówimy?

Omówimy dwa rodzaje rozwiązań dla MDS. Jeśli odległości są odległościami euklidesowymi, klasyczny MDS daje łatwe rozwiązanie algebraiczne. W większości aplikacji MDS potrzebne są metody iteracyjne, ponieważ dopuszczają wiele typów danych i odległości. Korzystają z dwufazowego algorytmu optymalizacji, przesuwając punkty w przestrzeni MDS małymi krokami, trzymając dane lub ich transformacje jako stałe i na odwrót, aż do osiągnięcia zbieżności.
W przypadku większości modeli MDS nie można znaleźć najlepszego możliwego

rozwiązania X, po prostu rozwiązując układ równań. Warunki dla rozwiązań MDS są na ogół tak skomplikowane, że są one nieodłączne algebraicznie. Dlatego rozwiązania MDS muszą być iteracyjnie aproksymowane za pomocą inteligentnych procedur wyszukiwania (algorytmów), które redukują Stress[1], wielokrotnie przesuwając punkty do nowych lokalizacji i sukcesywnie skalując bliskości aż znajdą minimum Stressu. Algorytmy tego rodzaju nie są potrzebne, jeśli chce się założyć lub udowodnić, że dane odmienności δ_{ij} - prawdopodobnie wyprowadzone z odwracania danych o podobieństwie - są odległościami euklidesowymi. W takim przypadku można użyć klasycznego MDS do analitycznego znalezienia rozwiązania MDS X.

2 Klasyczny MDS

2.1 Opis algorytmu klasycznego

Klasyczny MDS jest znany również jako skalowanie Torgersona i skalowanie Torgersona-Gowera. Działa w następujący sposób:

1. Podnosimy do kwadratu kolejne wartości wyjściowej macierzy: $\Delta^{(2)}$.
2. Tworzymy macierz $B_\Delta := -\frac{1}{2}\mathbf{Z}\Delta^{(2)}\mathbf{Z}$, gdzie $\mathbf{Z} = \mathbf{I} - \frac{\mathbf{11}'}{n}$.
3. Następnie diagonalizujemy macierz $B_\Delta = Q\Lambda Q'$.
4. Z macierzy wartości własnych Λ wybieramy wartości większe od zera (ozn. Λ_+) oraz dobieramy wektory własne dla tych wartości (ozn. Q_+).
5. Wynikiem klasycznego algorytmu MDS jest $\mathbf{X} = Q_+\Lambda_+^{\frac{1}{2}}$.

2.2 Przykład wykorzystania

Mamy wyjściową macierz oraz macierz kwadratów kolejnych wartości tej macierzy:

$$\mathbf{\Delta} = \begin{pmatrix} 0.00 & 4.05 & 8.25 & 5.57 \\ 4.05 & 0.00 & 2.54 & 2.69 \\ 8.25 & 2.54 & 0.00 & 2.11 \\ 5.57 & 2.69 & 2.11 & 0.00 \end{pmatrix} \qquad \mathbf{\Delta^{(2)}} = \begin{pmatrix} 0.00 & 16.40 & 68.06 & 31.02 \\ 16.40 & 0.00 & 6.45 & 7.24 \\ 68.06 & 6.45 & 0.00 & 4.45 \\ 31.02 & 7.24 & 4.45 & 0.00 \end{pmatrix}.$$

[1] $\text{Stress} - 1 = \sqrt{\sum_{i<j}\left(d_{ij}(\mathbf{X}) - \widehat{d}_{ij}\right)^2 / \sum_{i<j} d_{ij}^2(\mathbf{X})}$, gdzie $d_{ij}(\mathbf{X})$ to odległości rzeczywiste, a $\widehat{d}_{ij}$ to ich przybliżenia

Tworzymy macierz B_Δ zgodnie ze wzorem powyżej.

$$\mathbf{Z} = \begin{pmatrix} 0.75 & -0.25 & -0.25 & -0.25 \\ -0.25 & 0.75 & -0.25 & -0.25 \\ -0.25 & -0.25 & 0.75 & -0.25 \\ -0.25 & -0.25 & -0.25 & 0.75 \end{pmatrix} \qquad \mathbf{B_\Delta} = \begin{pmatrix} 20.52 & 1.64 & -18.08 & -4.09 \\ 1.64 & -0.83 & 2.05 & -2.87 \\ -18.08 & 2.05 & 11.39 & 4.63 \\ -4.09 & -2.87 & 4.63 & 2.33 \end{pmatrix}$$

Trzecim krokiem jest diagonalizacja obliczonej powyżej macierzy B_Δ, której macierzą wartości własnych oraz wektorów własnych są odpowiednio

$$\mathbf{\Lambda} = \begin{pmatrix} 35.71 & 0.00 & 0.00 & 0.00 \\ 0.00 & 3.27 & 0.00 & 0.00 \\ 0.00 & 0.00 & -0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -5.57 \end{pmatrix}, \qquad \mathbf{Q} = \begin{pmatrix} 0.77 & 0.04 & -0.50 & 0.39 \\ 0.01 & -0.61 & -0.50 & -0.61 \\ -0.61 & -0.19 & -0.50 & 0.59 \\ -0.18 & 0.76 & -0.50 & -0.37 \end{pmatrix}.$$

Przejdźmy od razu do obliczenia wyniku klasycznego algorytmu MDS dla tych danych.

$$\mathbf{X} = Q_+ \Lambda_+^{\frac{1}{2}} = \begin{pmatrix} 0.77 & 0.04 \\ 0.01 & -0.61 \\ -0.61 & -0.19 \\ -0.18 & 0.76 \end{pmatrix} \begin{pmatrix} 35.71 & 0.00 \\ 0.00 & 3.27 \end{pmatrix}^{\frac{1}{2}}$$

$$= \begin{pmatrix} 0.77 & 0.04 \\ 0.01 & -0.61 \\ -0.61 & -0.19 \\ -0.18 & 0.76 \end{pmatrix} \begin{pmatrix} 5.98 & 0.00 \\ 0.00 & 1.81 \end{pmatrix} = \begin{pmatrix} 4.62 & 0.07 \\ 0.09 & -1.11 \\ -3.63 & -0.34 \\ -1.08 & 1.38 \end{pmatrix}$$

W celu sprawdzenia jakości otrzymanego rozwiązania, porównujemy odległości otrzymanych punktów (macierz D; odległości pomiędzy punktami z Z) z początkowymi odległościami (macierz Δ). W wyniku otrzymujemy:

$$\mathbf{D} = \begin{pmatrix} 0 & 4.68 & 8.26 & 5.85 \\ 4.68 & 0 & 3.8 & 2.75 \\ 8.26 & 3.8 & 0 & 3.08 \\ 5.85 & 2.75 & 3.08 & 0 \end{pmatrix} \qquad \mathbf{\Delta - D} = \begin{pmatrix} 0 & -0.63 & -0.01 & -0.28 \\ -0.63 & 0 & -1.26 & -0.06 \\ -0.01 & -1.26 & 0 & -0.97 \\ -0.28 & -0.06 & -0.97 & 0 \end{pmatrix}.$$

W tym przykładzie otrzymane odległości między punktami są tylko w przybliżeniu równe podanym wcześniej danym. Spowodowane jest to tym, że różnice w Δ nie są odległościami euklidesowymi, jak zakłada klasyczne MDS. Matematycy zauważyliby, że w trzecim kroku, ponieważ różnice powinny być odległościami euklidesowymi to wartości własne powinny być nieujemne. Jeśli wystąpią ujemne wartości własne, można określić je jako "błąd" w różnicach, pod warunkiem, że te ujemne wartości własne są względnie małe. W powyższym przykładzie założenie to

wydaje się jednak trudne do uzasadnienia, ponieważ jedna ujemna wartość własna (= -5,57) jest dość duża.
Dlaczego ktoś chciałby założyć, że dane dotyczące odmienności są odległościami euklidesowymi (z wyjątkiem elementu błędu)? Uzasadnienie musi pochodzić z tego, że dane są generowane lub gromadzone. Jeśli osoby zostaną poproszone bezpośrednio o ocenę odmienności parami, wówczas możliwe jest postawienie hipotezy, że obserwowane reakcje numeryczne są co najmniej wielkościami zbliżonymi do odległości.
Korelacje jednak zdecydowanie nie są odległościami euklidesowymi, lecz raczej produktami skalarnymi *z założenia*. Dlatego w tym przypadku należy pominąć kroki 1 i 2, i rozpocząć bezpośrednio od kroku 3. Oznacza to wykonanie analizy składowej głównej. Alternatywnym podejściem jest najpierw przekonwertować skalarne produkty na odległości. W przypadku korelacji ta konwersja wynosi $d_{ij} = \sqrt{2 - 2r_{ij}}$.

2.3 Podsumowanie działania klasycznego MDS

W przypadku większych błędów (jak w pierwszym przykładzie) klasyczny MDS szybko osiąga swoje granice jako użyteczna metoda. Generuje to najlepsze możliwe rozwiązanie, ale minimalizuje to kryterium znane jako Strain, które nie jest tak łatwo interpretowalne jak Stress. Co więcej, zazwyczaj dane są co najwyżej na poziomie skali interwałowej. Dlatego nie należy interpretować danych bezpośrednio jako odległości, ale raczej pozwolić na optymalne ponowne skalowanie podczas mapowania ich na odległości.

3 Iteracyjny MDS

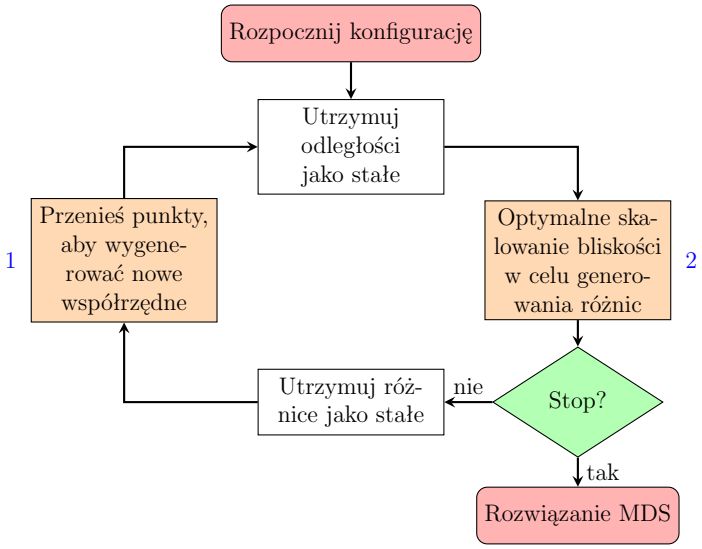
3.1 Opis algorytmu iteracyjnego

Iteracyjne algorytmy MDS działają w dwóch fazach. W każdej fazie pierwszy zestaw parametrów (odległości lub różnice, odpowiednio) przyjmuje się jako wartości stałe, natomiast drugi zestaw argumentów jest modyfikowany w taki sposób, aby zmniejszyć Stress:

1. Różnice są stałe; punkty w przestrzeni MDS są przesuwane (tzn. X_t zmienia się na X_{t+1}), tak aby odległości X_{t+1} minimalizowały funkcję Stress.

2. Konfiguracja MDS jest stała; różnice są skalowane w granicach ich poziomu skali, tak aby funkcja Stress była zminimalizowana (optymalne skalowanie).

Jeśli po t krokach proces nie zmniejszy wartości Stress o więcej niż pewna ustalona wartość (np. 0.0005), algorytm zostanie zatrzymany, a X_t będzie optymalnym rozwiązaniem.



Rysunek 1: Reguła iteracyjnego algorytmu MDS

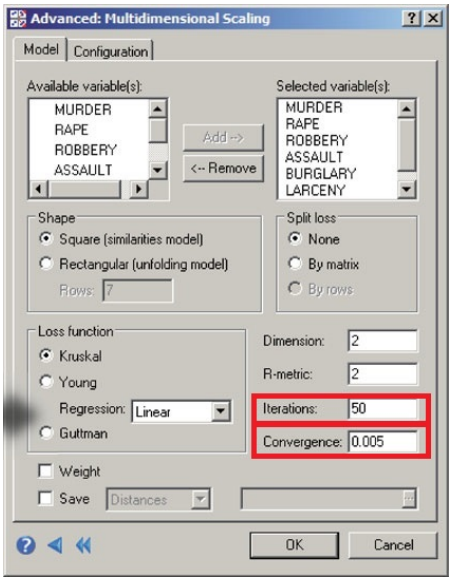
3.2 Problemy

Faza 1 równa się trudnemu problemowi matematycznemu o $n \cdot m$ nieznanych parametrach, wartości X. Aby go rozwiązać, zastosowano różne algorytmy optymalizacji. Obecnie najlepszym algorytmem jest procedura SMACOF (De Leeuw i Heiser 1980; Borg and Groenen 2005), ponieważ gwarantuje to w praktycznych sytuacjach, że iteracje zbiegną przynajmniej do lokalnego minimum Stresu. Inne kryterium może również służyć do oceny jakości algorytmów MDS (Basalaj 2001). Faza 2 stwarza stosunkowo łatwy problem. W interwałowym MDS problem rozwiązuje się poprzez regresję liniową. Znajduje dodatnie i multiplikatywne współczynniki, które liniowo przekształcają bliskości w różnice takie, że Stress jest minimalizowany dla danych odległości. W przypadku innych modeli MDS dostępne są również odpowiednie procedury regresji.
Są to jednak kwestie czysto matematyczne. Użytkownicy MDS nie muszą się nimi przejmować. Powinni po prostu korzystać z programów MDS, tak jak kierowcy korzystają z samochodów: Kierowcy muszą wiedzieć jak prowadzić samochód, ale nie muszą rozumieć fizyki silników spalinowych. Kierowcy powinni jednak wiedzieć, jak uruchomić samochód (np. upewnić się, że ma wystarczającą ilość paliwa), a użytkownicy MDS muszą prawidłowo wprowadzić dane do programu i ustawić odpowiednie opcje, aby dotrzeć tam, gdzie chcą, czyli uzyskać optymalne rozwiązania.

3.3 Opcje programów

Ważny jest wybór dobrej początkowej konfiguracji. Wszystkie programy MDS oferują kilka alternatyw, które użytkownicy mogą wypróbować, aby sprawdzić, czy wszystkie doprowadzą do tego samego rozwiązania. PROSCAL pozwala na przykład powtórzyć proces MDS poprzez wiele różnych losowych konfiguracji początkowych lub wybrać jedną konkretną racjonalną konfigurację początkową (np. taką, która wynika z używania klasycznego MDS), lub użyć zewnętrznej, skonstruowanej przez użytkownika konfiguracji początkowej.
Zaleca się, aby zawsze wpływać na wybór początkowej konfiguracji zamiast pozostawić ten wybór programowi MDS. Często dobrym wyborem jest użycie konfiguracji początkowej zbudowanej na podstawach merytoryczno-teoretycznych.

W zależności od konkretnego programu MDS, różne "techniczne" opcje oferowane są użytkownikom MDS. Te opcje mogą silnie wpłynąć na ostateczne rozwiązanie MDS, ponieważ często zmuszają algorytm do zakończenia iteracji, choć Stress można jeszcze poprawić. W oknie GUI programu SYSTAT użytkownik może ustawić maksymalną liczbę iteracji i zdefiniować liczbowe kryterium zbieżności. Ze względów historycznych (tj. aby zaoszczędzić czas i koszty), domyślne wartości dla tych parametrów są bardzo często ustawione we wszystkich programach MDS zbyt asekuracyjnie, aby iteracje zostały zakończone zbyt wcześnie. Użytkownicy powinni ustawić te parametry w taki sposób, aby program mógł wykonać tyle iteracji ile jest koniecznych, aby zmniejszyć Stress. Czas obliczania nie jest problemem w nowoczesnych programach MDS.



Rysunek 2: Okno GUI programu SYSTAT

4 Podsumowanie

Jeśli dane są odległościami euklidesowymi (poza błędami), klasyczny MDS jest wygodną metodą algebraiczną do wykonywania MDS. Konwertuje dane na produkty skalarne, a następnie odnajduje konfigurację MDS za pomocą diagonalizacji. Algorytmy iteracyjne są za to bardziej elastyczne: umożliwiają optymalne ponowne skalowanie danych i różne odmiany odległości Minkowskiego, a nie tylko odległości

euklidesowe. Takie programy zaczynają się od zdefiniowania lub użycia konfiguracji początkowej, a następnie modyfikują ją poprzez małe przesuwanie punktów, aby zmniejszyć Stress. Odległości tej konfiguracji są następnie wykorzystywane jako cele dla optymalnego ponownego skalowania danych (w ten sposób generując różnice) w granicach poziomu skali danych. Ten proces modyfikacji konfiguracji MDS (ze stałymi różnicami) i ponownego skalowania różnic (ze stałymi odległościami) jest powtarzany aż do optymalnego rozwiązania. Obecnie najlepszym algorytmem przesuwania punktów jest SMACOF (ponowne skalowanie danych jest wykonywane poprzez regresję).

Literatura

[1] Ingwer Borg, Patrick J.F. Groenen, Patrick Mair: Applied Multidimensional Scaling, Springer - Verlag Berlin Heidelberg, 2013.