Dokumentacja

Aplikacja do analizy danych pomiarowych umożliwiająca wykrywanie wartości nietypowych.

Prowadzący: dr inż. Grzegorz Kołaczek

Autorzy:
1. Adrian Zybała, nr indeksu: 179206
2. Rafał Maliszewski, nr indeksu: 179292
3. Paweł Plicner, nr indeksu: 179145
4. Paweł Łabuda, nr indeksu: 179211

Spis treści

Cel przedsięwzięcia

Celem przedsięwzięcia jest zbudowanie aplikacji, której zadaniem będzie wykrywanie anomalii w monitorowanym systemie oraz zbadanie użytych metod w kontekście ich skuteczności. Aplikacja powinna wykrywać anomalie w zbiorach danych, które nie zmieniają się jak również dla danych dodawanych do zbioru w czasie rzeczywistym. Celem aplikacji nie jest ekstrakcja badanych cech, a jedynie ich przetwarzanie.

Wykrywanie anomalii

Wykrywanie anomalii jest zadaniem rozpoznawania wzorców (ang. pattern recognition task), którego celem jest znajdowanie nienormalnych lub nieznanych zachowań w badanym systemie. W rozpoznawaniu wzorców dąży się do klasyfikacji danych w oparciu o wiedzę aprioryczną lub informacje uzyskane przy pomocy analizy statystycznej badanych zbiorów danych. Klasyfikowane obiekty to najczęściej wyniki pomiarów w postaci wektora cech, na podstawie których klasyfikujemy obiekt.

Metody wykrywania anomalii

Do badania anomalii wykorzystuje się wiele metod oraz struktur, które mają swoje odmiany, czy po prostu zmienne parametry. W naszej aplikacji skupiliśmy się na sześciu z nich, których parametrami można manipulować, a są to:

1. Sieć neuronowa Back Propagation,
2. Samoorganizująca sieć Kohonena,
3. Sieć bayesowska,
4. Naiwny klasyfikator bayesowski,
5. Lokalny czynnik wartości oddalonej (ang. local outlier factor)
6. Algorytm k-najbliższych sąsiadów

Sieć neuronowa Back Propagation

Sieć neuronowa jest to struktura wzajemnie połączonych elementów nazywanych neuronami. W projekcie wykorzystaliśmy klasyczny model neuronu. Do każdego połączenia przypisana jest waga w_i. Na wejście pojedynczego neuronu podawany jest wektor $[x] = [x_1, x_2, \dots, x_n]^T$. Wówczas neuron oblicza całkowite pobudzenie $net = \sum_j x_j w_j$. Po obliczeniu pobudzenia neuronu jest ono zamieniane na wartość wyjścia poprzez przetworzenie jej przez funkcję aktywacji. Po obliczeniu wyjść w warstwie przetwarzanie jest przenoszone do warstwy wyższej. Faza ta nazywana jest fazą przesłania (propagacji) do przodu.

Celem uczenia metodą propagacji wstecznej jest minimalizacja błędu między wektorem wyjściowym $[y] = [y_1, y_2, \dots, y_m]^T$ a wektorem $[o] = [o_1, o_2, \dots, o_m]^T$ otrzymanym na wyjściu sieci. Minimalizacja ta dotyczy sumy błędów kwadratowych pojawiających się na wyjściu. Dla p-tego wzorca:

$$E_p = \frac{1}{2}\sum_k \left(y_{pk} - o_{pk}\right)^2$$

gdzie k jest kolejnym neuronem w warstwie wyjściowej.
Będziemy zmieniać wartość wagi j w neuronie o mały przyrost zgodnie ze wzorem:
$w_j(t+1) = w_j + \Delta w_j$

W warstwie wyjściowej:

$$\Delta w_j = \Delta_p w_{kj}^{wy} = \alpha\left(y_{pk} - o_{pk}\right) f_k'^{wy}\left(net_{pk}^{wy}\right) i_{pj}$$

gdzie i_{pj} jest wartością wyjściową neuronu w warstwie ukrytej, z którym neuron wyjściowy powiązany jest połączeniem j.

W warstwach ukrytych:

$$\Delta w_j = \Delta_p w_{ji}^{u} = \eta f_j'^{u}\left(net_{pj}^{u}\right) x_{pi} \sum_k \left(y_{pk} - o_{pk}\right) f_k'^{wy}\left(net_k^{wy}\right) w_{kj}^{wy}$$

gdzie x_{pi} jest wartością wyjściową neuronu w poprzedzającej warstwie ukrytej, z którym neuron powiązany jest połączeniem i. Współczynniki α i η są współczynnikami uczenia.

Najpierw obliczane są wagi w warstwie wyjściowej, następnie w warstwach ukrytych, a dopiero na koniec po tych obliczeniach wagi są uaktualniane. Faza ta nazywa się fazą przesłania wstecz.

Każdy rekord danych jest pojedynczym wektorem wejściowym, natomiast za decyzję, czy występuje anomalia odpowiada pojedynczy, binarny neuron wyjściowy.

Samoorganizująca sieć Kohonena

Samoorganizująca sieć Kohonena (Self-Organizing Map) jest bardzo często wykorzystywaną siecią neuronową. Znajduje ona zastosowanie w wielu problemach, także podczas wykrywania wartości nietypowych w zbiorach danych. Zaletą sieci tej jest to, że jest uczona w sposób nienadzorowany. To znaczy, że nie zbiór danych uczących nie musi być zbiorem opisanym. W przypadku wykrywania anomalii należy sieć tą wyuczyć przy pomocy zbioru zawierającego wyłącznie dane poprawne, nie zawierające anomalii. W sieci tej wyróżnia się dwie warstwy: warstwę wejściową oraz warstwę wyjściową będąca warstwą rywalizacji. Warstwa rywalizacji zwana jest także warstwą Kohonena. Neurony w niej leżą na płaszczyźnie, która stanowi mapę odwzorowującą wektory wejściowe podawane sieci podczas uczenia. Jeżeli na przykład większość wzorców uczących będzie miała szczególną charakterystykę, odpowiednia liczba wektorów wag będzie miała podobną cechę. Każdy neuron należący do warstwy wyjściowej posiada przypisany wektor wag, tego samego wymiaru co liczba neuronów w warstwie wejściowej.

Wagi sieci SOM korygowane są z wykorzystaniem metody WTM (*Winner Takes Most*). Dzięki temu uzyskujemy klasteryzację danych wejściowych. W celu uzyskania lepszych wyników klasteryzacji, podczas uczenia sieci wykorzystana została funkcja Gaussa określająca stopień zmiany wartości wag neuronu zwycięzcy (*ang. BMU - Best Matching Unit*) oraz jego sąsiedztwa. Reguła zmiany wag:

$$[w(t+1)]_i = \begin{cases} [w(t)]_i + \alpha(t) f_{Nc}(t)([x] - [w(t)]_i) & i \in N \\ [w(t)]_i & w\ przeciwnym\ wypadku \end{cases}$$

W powyższym wzorze $[x]$ jest wektorem wejściowym, f_{Nc} to funkcja sąsiedztwa (w naszym przypadku funkcja Gaussa), zaś α jest współczynnikiem uczenia. Zastosowana funkcja Gaussa:

$$f_{Nc} = \exp\left(\frac{-d^2([x],[w])}{2R^2}\right),$$

gdzie R to promień sąsiedztwa, d – odległość między wektorem wejściowym $[x]$ a wektorem wag $[w]$ neuronu.

Gdy sieć zostanie wyuczona, neurony warstwy konkurencyjnej stanowią odwzorowanie

danych uczących. Wektory wag nazywane są nieraz prototypami reprezentowanych klas wektorów wejściowych, które najlepiej reprezentują w zbiorze uczącym. Zwycięski neuron wyznaczany jest przy pomocy odległości euklidesowej wektora wag od wektora wejściowego. Zwycięzcą zostaje ten, dla którego odległość ta jest najmniejsza. Podobne wektory wejściowe aktywują więc ten sam neuron. W miarę kolejnych iteracji uczenia sieci, promień sąsiedztwa oraz współczynnik uczenia maleją zgodnie z regułami:

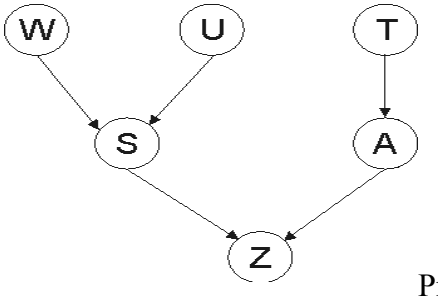
$$R(t) = R_{max}\left(\frac{R_{min}}{R_{max}}\right)^{\frac{t}{t_{max}}}$$
$$\alpha(t) = \alpha_{max}\left(\frac{\alpha_{min}}{\alpha_{max}}\right)^{\frac{t}{t_{max}}},$$

gdzie R_{max} oraz R_{min} to maksymalny i minimalny promień sąsiedztwa, t to aktualna iteracja algorytmu, zaś t_{max} określa maksymalną liczbę kroków algorytmu.

Sieć SOM uczona z wykorzystaniem zwyczajnych, nie odbiegających od normy danych wejściowych odwzorowuje je w wagach neuronów warstwy Kohonena. Wszelakie odstępstwa od normy, czy anomalie powinny w znacznie gorszym stopniu pasować do stworzonych klastrów oraz ich prototypów. Właśnie ta cecha wykorzystywana jest do wykrywania anomalii. Błąd dopasowania danych wejściowych do wzorców wyuczonej sieci samoorganizującej nazywany jest błędem kwantyzacji. To na jego podstawie możliwe staje się podjęcie decyzji o występowaniu wartości nietypowych w obserwowanych danych. Dla wyuczonej sieci Kohonena wyznaczany jest średni, maksymalny oraz minimalny błąd kwantyzacji dla danych „normalnych". Następnie podczas właściwej pracy sieci neuronowej, dla każdego podanego wektora wejściowego badany jest błąd kwantyzacji, który to porównywany jest z wartościami błędów kwantyzacji dla danych uczących. Trudnością w tym momencie jest ustalenie odpowiedniego progu wartości błędu kwantyzacji, przy którym stwierdzane zostało by wykrycie wartości nietypowej. Próg ten powinien być oczywiście większy od maksymalnego błędu kwantyzacji zaobserwowanego dla zbioru uczącego po wytrenowaniu sieci neuronowej.

Sieć bayesowska

Sieć bayesowska to struktura określająca zależność zmiennych losowych(zdarzeń) od siebie. Formalnie sieć jest zdefiniowana jako skierowany graf acykliczny, w którym wierzchołki reprezentują zmienne losowe(zdarzenia), a łuki związki przyczynowe między zdarzeniami. Jeśli rodzicem zdarzenia A jest zdarzenie B, to rozkładem zmiennej A jest rozkład warunkowy P(A|B). Jeśli rodziców jest więcej np. $B_1,...,B_n$ to rozkład zmiennej A jest definiowany jako rozkład warunkowy $P(A|B_1,...,B_n)$.
Prawdopodobieństwo całkowite sieci bayesowskiej jest iloczynem wszystkich prawdopodobieństw w wierzchołkach.



Prawdopodobieństwo całkowite na lewym obrazku wynosi:
P(W,U,T,S,A,Z) = P(W)*P(U)*P(T)*P(S|W,U)*P(A|T)*P(Z|S,A).
Dzięki temu wzorowi jesteśmy w stanie stwierdzić, po estymacji każdego rozkładu w węźle, jakie

jest prawdopodobieństwo, że punkt jest anomalią czy też nie.

Naiwny klasyfikator bayesowski

Naiwny klasyfikator bayesowski (ang. naive bayes classifier) to prosty klasyfikator należący do rodziny klasyfikatorów probabilistycznych. Swoje działanie opiera na założeniu o wzajemnej niezależności predykatów. Klasyfikator określany jest mianem naiwnego, gdyż w świecie rzeczywistym powyższe założenie jest najczęściej nieprawdziwe, jednakże pomimo funkcjonowania w oparciu o bardzo uproszczone założenia naiwne klasyfikatory bayesowskie znajdują wiele praktycznych zastosowań. Klasyfikator ten należy do algorytmów uczonych w sposób nadzorowany.

$$classify(f_1, \dots, f_n) = \operatorname*{argmax}_c p(C = c) \prod_{i=1}^{n} p(F_i = f_i | C = c)$$

Algorytm swoje działanie opiera na klasyfikatorze powstałym w wyniku połączenia naiwnego modelu probabilistycznego Bayesa z regułą decyzyjną, która ma za zadanie wybrać najbardziej prawdopodobną hipotezę. Opisywany klasyfikator jest zdefiniowany następująca funkcją classify: gdzie prawdopodobieństwo wystąpienia pewniej klasy opisane jest w następujący sposób:

$$P(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

$\mu_c -$ *średnia arytmetyczna wektora cech x klasy c*

$\sigma_c^2 -$ *wariancja wektora cech x klasy c*

W niektórych zadaniach klasyfikacji bardzo często można uzyskać lepsze rezultaty rozbijając rozpatrywanie zadanie na kilka niezależnych podzadań. Na przykład w przypadku wykrywania anomalii występujących w budowie ciała ludzkiego dane zadanie można rozbić na dwa odrębne podzadania wykrywania anomalii w budowie mężczyzn oraz kobiet.

Lokalny czynnik wartości oddalonej

Lokalny czynnik wartości oddalonej (ang. local outlier factor) to wartość wyliczana na podstawie wartości lokalnej osiągalności gęstości (ang. local reachability density) badanego punktu w porównaniu do k-najbliższych sąsiadów(punktów).
Lokalny czynnik wartości oddalonej jest zdefiniowany wzorem:

$$LOF_k(A) := \frac{\sum_{B \in N_k(A)} \frac{lrd(B)}{lrd(A)}}{|N_k(A)|} = \frac{\sum_{B \in N_k(A)} lrd(B)}{|N_k(A) * lrd(A)|}$$, gdzie

$N_k(A)$ – zbiór sąsiadów punktu A
lrd(X) – lokalna osiągalność gęstości, zdefiniowana wzorem:

$$lrd_k(A) := \left(\frac{\sum_{B \in N_k(A)} reachability - distance(A, B)}{|N_k(A)|} \right)^{-1}$$, gdzie

$reachability\text{-}distance_k(A,B) = \max(k\text{-}distance(B), d(A,B))$, gdzie
k-distance(B) to odległość punktu B do jego k-tego sąsiada, a d(A,B) to odległość euklidesowa punktów A i B.

Otrzymany współczynnik pozwala na zakwalifikowanie punktu do grupy anomalii lub nie. Otrzymana wartość LOF powinna być w okolicach 1.0 jeśli otrzymany punkt nie należy do grupy anomalii. Ciężko zdefiniować dozwolony przedział, więc metoda jest parametryzowana dopuszczalnym odchyleniem współczynnika jak również ilością najbliższych sąsiadów których powinno się wziąć pod uwagę.

Algorytm k-najbliższych sąsiadów

Algorytm k najbliższych sąsiadów (ang. k-nearest neighbor algorithm) jest metodą klasyfikacji obiektów na podstawie ich umiejscowienia w przestrzeni cech stworzonej z informacji uzyskanych z obiektów znajdujących się w zbiorze uczącym. Algorytm ten zaliczany jest do grona najprostszych algorytmów maszynowego uczenia, jednakże często znajduje on zastosowanie ze względu na swoją prostotę. Metoda k najbliższych sąsiadów jest użyteczna w zadaniach klasyfikacji w sytuacjach, gdy zależności występujące pomiędzy badanymi obiektami są złożone lub nietypowe, co wpływa negatywnie na możliwości ich zaklasyfikowania metodami klasycznymi. Opisywany algorytm należy do rodziny algorytmów uczonych w sposób nadzorowany. Oznacza to, że zbiór danych wejściowych (uczących) musi być zbiorem opisanym – każda obserwacja ze zbioru uczącego, opisana przy pomocy wektora cech, musi posiadać także etykietę objaśniającą.

Kroki algorytmu:
1. Oblicz odległość pomiędzy aktualnie przetwarzanym punktem ze zbioru uczącego a punktem będącym podmiotem klasyfikacji
2. Posortuj w kolejności malejącej punkty ze zbioru uczącego na podstawie wyliczonych odległości
3. Wybierz k (ustalony z góry parametr) najbliżej położonych punktów ze zbioru uczącego
4. Zwróć klasę (etykietę objaśniającą) obiektów najczęściej występujących w stworzonym podzbiorze

Do wyznaczenia odległości pomiędzy wybranymi punktami najczęściej używana jest metryka euklidesowa, jednakże użyta zostać może dowolna metryka. Parametr k to najczęściej liczba nieparzysta, co zmniejsza szanse otrzymania dwóch lub większej ilości obiektów, które posiadają tyle samo wystąpień w rozpatrywanym podzbiorze. Zwiększenie parametru k pozwala ograniczyć ilość szumu występującego w zadaniu klasyfikacji, jednakże wraz ze wzrostem parametru zmniejsza się także przestrzeń rozgraniczająca poszczególne klasy obiektów.

Naiwna wersja algorytmu jest bardzo łatwa do zaimplementowania, jednakże jest wymagająca pod względem obliczeniowym, szczególnie dla bardzo dużych zbiorów treningowych.

Co udało się zrobić?

- Działająca aplikacja,
- Rysowanie wykresów dla każdej zmiennej(cechy),
- Oznaczanie czerwoną pionową kreską wartości nietypowych,
- Analiza offline danych zapisanych w logach,
- Implementacja sieci neuronowej BP,
- Implementacja samoorganizującej sieci Kohonena,
- Implementacja sieci bayesowskiej,
- Implementacja Naiwnego klasyfikatora bayesowskiego,
- Implementacja lokalnego czynnika wartości oddalonej,
- Implementacja metody odległościowej(k-najbliższych sąsiadów),

Plany na przyszłość

- Obsługa bazy danych sql, czyli pobieranie i obsługa danych w czasie rzeczywistym,
- Edytor węzłów dla sieci bayesowskiej (wizualnie, typu drag&drop),
- Rozbudowa interfejsu użytkownika,
- Ponowna rewizja zaimplementowanych metod dla nowych danych m.in. przetestowanie sieci neuronowej uczonej metodą propagacji wstecznej (perceptron wielowarstwowy),
- Implementacja nowych metod,
- Automatyczna, półautomatyczna lub manualna parametryzacja metod,
- Dostęp zdalny przez przeglądarkę www, aplikację mobilną, klienta desktopowego.

Bibliografia

[1] Markowska-Kaczmar Urszula, Kwaśnicka Halina, *Sieci neuronowe w zastosowaniach*, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 2005

[2] Rewbenio A. Frota, Guilherme A. Barreto, C. M. Mota, *Anomaly Detection In Mobile Communication Networks Using The Self-Organizing Map,* Fortaleza, IOS Press, 2007

[3] http://en.wikipedia.org/wiki/Local_outlier_factor – Local Outlier Factor