Uniwersytet Mikołaja Kopernika
Wydział Matematyki i Informatyki

Krzysztof Dobosz
nr albumu 158052

Praca magisterska
na kierunku Informatyka

Statystyczne testy oceny istotności różnic w wynikach uzyskiwanych przez różne systemy uczące się

Praca wykonana pod kierunkiem
prof. dra hab. Włodzisława Ducha,
Katedra Informatyki Stosowanej

Toruń, 2006

Moim rodzicom

za życie, miłość i trud

włożony w moje wykształcenie

Spis treści

Rozdział 1

Wstęp

W uczeniu maszynowym istnieje już wiele metod, które dobrze radzą sobie z problemem klasyfikacji danych. Wielu ludzi zaangażowanych jest w tworzenie nowych, bardziej wyrafinowanych, a przez to dokładniejszych algorytmów. W wyniku tego powstaje duża liczba modeli klasyfikujących a razem z nią pojawia się naturalne pytanie: „Która spośród dostępnych metod jest najlepsza?". Oczywiście ciężko jest udzielić jednoznacznej odpowiedzi na tak szeroko postawione pytanie. Niektóre algorytmy lepiej radzą sobie na jednych danych inne na innych, a różnice w działaniu na poszczególnych zbiorach są często olbrzymie. Z tego też powodu jesteśmy zmuszeni stawiać bardziej konkretne pytanie: „Mając pewien zbiór (zbiory) danych, pytamy, który model radzi sobie lepiej z problemem klasyfikacji na tym zbiorze?".

Jednak i tutaj pojawia się pewien problem: co oznacza, że dany model jest lepszy od innego? Możemy twierdzić, iż lepszy to ten, który zrobi mniej błędów na zbiorze testowym. Ale czy to wystarczy? Czy możemy wówczas powiedzieć, iż ten model jest generalnie lepszym klasyfikatorem na tego typu danych? A może to tylko przypadek sprawił, że owe metody wygenerowały takie prawdopodobieństwa popełnienia błędów? Wiele przecież zależy od tego jakie ustawimy parametry oraz jaka część danych posłuży jako dane treningowe, a jaka jako dane testowe. Z drugiej strony w uczeniu maszynowym danych nigdy nie mamy za wiele i nawet tak wyrafinowane metody resamplingu danych jak kroswalidacja nie są w stanie zagwarantować

nam pewności, że zaobserwowane różnice w błędach klasyfikatorów nie są wyłącznie dziełem przypadku. Musimy mieć zatem jakiś poziom zaufania do podejmowanej decyzji – poziom istotności różnicy pomiędzy testowanymi algorytmami. I tu z pomocą przychodzą statystyczne testy istotności.

Struktura niniejszej pracy przedstawia się w następujący sposób. Rozdział 2. obrazuje ogólny schemat przeprowadzania statystycznych testów istotności wraz z wyróżnieniem pewnych metod Inteligencji Obliczeniowej, pozwalających na tworzenie prób z wynikami modeli klasyfikujących. W Rozdziale 3. zaprezentowana została dokładna charakterystyka testów, a w niej trzy kryteria podziału zbioru testów, jak również szczegółowe omówienie wybranych testów istotności tak dla prób sparowanych jak i niesparowanych. Następnie w Rozdziale 4. opisany został prototypowy plugin do wykonywania testów statystycznych w powstającym nowym systemie do inteligentnej analizy danych o nazwie Intemi. Na zakończenie, w Rozdziale 5., przedstawiona zostanie krótka analiza porównawcza wybranych testów istotności.

Podziękowania

Chciałbym wyrazić serdeczne podziękowania mojemu promotorowi Panu Profesorowi Włodzisławowi Duchowi za udzieloną pomoc oraz kierownictwo nad pracą, jak również głównym projektantom systemu Intemi Panu Doktorowi Krzysztofowi Grąbczewskiemu i Panu Doktorowi Norbertowi Jankowskiemu za wiele cennych uwag zarówno odnośnie treści pracy jak i implementacji pluginu.

Rozdział 2

Ogólny schemat przeprowadzania testów istotności

2.1. Zbieranie wyników

Pomimo, iż celem pracy nie jest omawianie sposobów zbierania wyników osiąganych przez metody uczenia maszynowego, a jedynie przegląd testów statystycznych mających zastosowanie w porównywaniu różnic pomiędzy systemami uczącymi się, to jednak dużo problemów związanych z samym testowaniem istotności różnic pojawia się już na tym etapie. Z tego powodu w tym podrozdziale zostaną przedstawione pokrótce zarówno sposoby tworzenia wektorów wyników dla porównywanych algorytmów, jak również związane z nimi problemy naruszające podstawowe założenia testów statystycznych.

W celu przeprowadzenia statystycznego testu istotności musimy posiadać wyniki pewnej próby losowej, a więc zaobserwowane wartości badanej cechy elementów populacji generalnej, które zostały wybrane w sposób losowy do próby.

W przypadku porównywania metod uczenia maszynowego badaną cechą może być przykładowo dokładność klasyfikacji wektorów uzyskiwana na zbiorach testowych lub też prawdopodobieństwo popełnienia błędu podczas tej klasyfikacji. W obu przypadkach wartości wektora wyników próby będą liczbami rzeczywistymi z przedziału [0, 1]. Naturalnym jest, iż w celu

utworzenia wektora wyników długości n, potrzebujemy n pomiarów dokładności klasyfikacji, a więc zarówno n zbiorów treningowych – do wyprodukowania takiej liczby modeli klasyfikujących, jak również n zbiorów testowych – do uzyskania tylu pomiarów dokładności klasyfikacji. Każdy algorytm bowiem musi być trenowany i testowany na innym zbiorze testowym w celu zapewnienia niezależności pomiędzy poszczególnymi wynikami, co jest jednym z ważniejszych założeń testów operujących na tego typu próbach.

Z powodu jednego z największych problemów uczenia maszynowego, mianowicie problemu małej ilości danych, do utworzenia odpowiedniej liczby zbiorów treningowych oraz testowych możemy posłużyć się jedną z szeroko stosowanych w uczeniu maszynowym metod wielokrotnego losowania, czyli tak zwanych metod resamplingu danych. Najprościej mówiąc, metody resamplingu służą sztucznemu zwiększaniu ilości posiadanych danych. Oczywiście samych danych nie przybywa – nie tworzymy nowych wektorów – jedynie lepiej wykorzystujemy te, które posiadamy. Zamiast bowiem dzielić zbiór danych na dwa podzbiory (treningowy i testowy) oraz przeprowadzić w ten sposób tylko jedną kombinację trening + test, możemy na przykład podzielić go na k podzbiorów (gdzie k jest mniejsze od liczby wektorów w całym zbiorze) i traktować je jako osobne zbiory testowe. W takim przypadku uczenie (trening) przed każdym testem wykonuje się na pozostałych wektorach, tj. wektorach nie należących zbioru testowego w danej iteracji tego algorytmu. Opisana procedura nosi nazwę k-krotnej kroswalidacji.

Należy jednak zwrócić uwagę na fakt, iż generowanie zbiorów testowych i treningowych za pomocą metod resamplingu danych nie zapewnia całkowitej niezależności wyników uzyskiwanych na nich przez systemy uczące się. O ile zbiory testowe są rozłączne, o tyle zbiory treningowe nakładają się. I tak w przypadku 10-krotnej kroswalidacji dla każdych dwóch iteracji tego algorytmu część wspólną zbiorów treningowych stanowi $\frac{8}{9}$ każdego ze zbiorów.

Oprócz kroswalidacji najczęściej stosowanymi metodami resamplingu są leave-one-out, będąca w rzeczywistości specyficzną odmianą kroswalidacji (gdy wspomniane k jest równe liczbie wektorów treningowych) oraz bootstrapping. Poszczególne metody nie będą jednak dokładnie omawiane, ponieważ wykracza to poza ramy niniejszej pracy. Szczegółowe informacje na ich temat można znaleźć w [3, 17, 20, 21].

Jeśli wektor wyników uzyskany za pomocą pomiaru dokładności klasyfikacji (lub błędu klasyfikacji) będzie miał rozkład normalny, to wówczas możemy wykorzystać mocne testy parametryczne, z których najbardziej znanym jest z pewnością test t-Studenta (omówiony w dalszej części pracy). W przeciwnym wypadku możemy użyć jednego z testów dla danych porządkowych należącego do dużo liczniejszej grupy testów nieparametrycznych.

Kolejną cechą populacji, jaka może zostać poddana badaniu, jest poprawność klasyfikacji poszczególnych wektorów. Możemy na przykład liczbą 0 oznaczyć błędną decyzję podczas klasyfikacji wektora ze zbioru testowego oraz liczbą 1 decyzję poprawną, czyli przyjąć, że wartości komórek wektora wyników próby losowej będą należeć do zbioru {0, 1}. W takim podejściu wektor wyników będzie wektorem o rozkładzie dwumianowym, co pozwala na wykorzystanie do pomiaru istotności różnic testów dla danych nominalnych (skategoryzowanych) takich jak test McNemara, różnicy dwóch proporcji czy testu χ^2.

W tym rozwiązaniu wykorzystuje się jedynie jeden zbiór treningowy oraz jeden zbiór testowy. Wielkość próby losowej jest wówczas równa liczbie wektorów przydzielonych do zbioru testowego. Jak nietrudno zauważyć nie uwzględnia się w takim przypadku wariancji spowodowanej wyborem zbioru treningowego oraz testowego, jak i wewnętrznej wariancji algorytmów (tworzony jest tylko jeden klasyfikator). Jest to dość poważny mankament

testów operujących na tego typu próbach. Oczywiście problem ten nie dotyczy wspomnianych wcześniej metod resamplingu danych, w których zarówno tworzone są różne zbiory testowe i treningowe, jak również różne modele klasyfikujące.

W badaniu istotności różnic systemów uczących się, należy wziąć pod uwagę, iż wszelkie eksperymenty, jakie przeprowadzamy, są obarczone pewną niedoskonałością. Przykładowo losowy wybór zbiorów testowych i treningowych pociąga za sobą pewną zmienność eksperymentów. Dietterich [2] wyróżnia cztery główne źródła zmienności, które powinno się uwzględniać podczas porównywania algorytmów uczenia maszynowego:

1. Wybór zbioru testowego. Klasyfikatory mogą uzyskiwać różniące się wyniki na wybranym zbiorze testowym, podczas gdy na całej populacji wektorów mogą zachowywać się identycznie. Ten problem szczególnie dotyczy małych zbiorów danych.
2. Wybór zbioru treningowego. Pewne algorytmy nazywa się niestabilnymi z powodu dużej wrażliwości na różnice występujące w danych treningowych. Oczywiście niestabilność metody pociąga niestabilność uzyskiwanych wyników. Kuncheva [3] wymienia dwa przykłady takich algorytmów: drzewa decyzyjne oraz niektóre sieci neuronowe.
3. Wewnętrzna wariancja algorytmu. W niektórych systemach dużą rolę odgrywa losowość wewnętrznych elementów. Przykładowo sieć neuronowa powstała przy użyciu algorytmu wstecznej propagacji jest mocno zależna od wyboru parametrów inicjalizacyjnych. Innym przykładem mogą być algorytmy genetyczne, w których proces tworzenia klasyfikatorów oparty jest o losowość krzyżowania chromosomów. W ten sposób tworzone klasyfikatory mogą być znacząco różne pomimo trenowania ich na tych samych zbiorach danych.

4. Błąd przypadkowej klasyfikacji. W zbiorze testowym mogą znajdować się wektory niewłaściwie zaetykietowane, co wpływa na wyniki uzyskiwane przez wszystkie modele klasyfikujące.

W związku z powyższym Kuncheva [3] jak i Dietterich [2] zalecają stosowanie wielokrotnych treningów i testów algorytmów, w celu zmniejszenia błędów pomiarów wynikających z wymienionych źródeł zmienności.

2.2. Stawianie hipotez

Test statystyczny często nazywany jest procedurą weryfikacji hipotez statystycznych, bowiem głównym zadaniem takiego testu jest sprawdzenie słuszności (dokładniej prawdopodobieństwa prawdziwości) postawionej hipotezy, czyli jakiegoś sądu (przypuszczenia) dotyczącego (rozkładu) populacji generalnej, bez zbadania jej całości.

Prawdziwość hipotezy weryfikujemy za pomocą wyników próby losowej, czyli w przypadku oceny istotności różnic systemów uczących się, za pomocą wektorów z wynikami uzyskanymi w procesie testowania tych systemów na pewnych zbiorach danych.

W każdym statystycznym teście istotności należy sformułować dokładnie dwie hipotezy:

- hipotezę zerową oznaczaną przez H_0, która roztacza pewien sąd, na ogół o równości parametrów czy też o postaci rozkładu populacji generalnej; jest to hipoteza bezpośrednio sprawdzana danym testem

oraz

- hipotezę alternatywną (niezerowa) oznaczaną przez H_1, będącą hipotezą przeciwstawną do hipotezy zerowej, którą przyjmujemy w razie odrzucenia hipotezy zerowej w procedurze weryfikacyjnej.

Podczas testowania istotności różnic w wynikach uzyskiwanych przez metody uczenia maszynowego stawiamy następujące dwie hipotezy:

- hipoteza zerowa H_0: porównywane metody uzyskują takie same wyniki na danym zbiorze, to znaczy wyniki, które nie różnią się istotnie – wszelkie różnice pomiędzy nimi są jedynie dziełem przypadku;
- hipoteza alternatywna H_1: metody uzyskują istotnie różne wyniki.

Ponadto hipoteza alternatywna może być dwojakiego rodzaju: ukierunkowana lub nieukierunkowana. W przypadku hipotezy ukierunkowanej twierdzimy, iż jeden z dwóch badanych algorytmów jest istotnie lepszy lub gorszy od drugiego, zaś w przypadku hipotezy nieukierunkowanej, jedynie że są one istotnie różne bez typowania kierunku tej różnicy. Oczywiście rozróżnienie pomiędzy tymi dwoma rodzajami hipotez nie jest tylko kwestią odpowiedniego nazewnictwa. Istnieje jeszcze jedna poważna rozbieżność pomiędzy nimi. Chodzi mianowicie o wielkość obszaru krytycznego wyznaczonego w rozkładzie odpowiedniej statystyki. Na ogół obszar ten jest dokładnie dwa razy większy dla hipotezy nieukierunkowanej. Wynika to z symetrycznego rozkładu statystyki testowej. Bywają jednak sytuacje, w których taka własność nie zachodzi. Przykładem może być rozkład statystyki stosowanej w teście dokładności Fishera, należącego do grupy tak zwanych testów permutacyjnych.

2.3. Wyliczanie statystyki

Każdy test istotności wymaga wyliczenia jakiejś statystyki, na podstawie której podejmujemy decyzję o przyjęciu lub odrzuceniu hipotezy zerowej na pewnym poziomie istotności. Poziom istotności zostanie omówiony dokładnie w kolejnym podrozdziale, a tymczasem przyjrzyjmy się czym jest statystyka. Jak podaje Greń [1] statystykę definiujemy w następujący sposób:

Def.: *„Statystyką z próby nazywamy zmienną losową będącą dowolną funkcją wyników próby losowej."*

Statystyką nazwiemy zatem każdą funkcję, która na podstawie wyników próbki losowej generuje jakąś wartość (zwykle rzeczywistą). Rolę statystyki może spełniać przykładowo średnia arytmetyczna wyników próby czy też standardowe odchylenie, jednak w praktyce weryfikacji hipotez stosuje się bardziej wyrafinowane statystyki, które wykorzystują więcej informacji zawartych w próbie.

W przypadku badania istotności różnic systemów uczących się, statystykami będą funkcje wyników zebranych w fazie testowania tych systemów na pewnych zbiorach danych (zbiorach testowych).

2.4. Poziom istotności oraz błąd typu I i II

Formalną definicję poziomu istotności, zwanego również progiem istotności, możemy sformułować następująco (Greń [1]):

Def.: *„Poziomem istotności nazywamy prawdopodobieństwo popełnienia błędu pierwszego rodzaju w postępowaniu testującym hipotezę."*

W teorii weryfikacji hipotez statystycznych mianem błędu pierwszego rodzaju określa się błąd polegający na odrzuceniu prawdziwej hipotezy zerowej. W literaturze pojęcie to występuje również pod takimi nazwami jak: błąd typu I, błąd odrzucenia czy też alfa-błąd. Poziom istotności jest zatem oszacowaniem prawdopodobieństwa odrzucenia hipotezy zerowej, która w rzeczywistości jest prawdziwa. Standardowo próg istotności oznaczamy grecką literą α.

Zazwyczaj w teście istotności ustala się z góry pewien poziom istotności, oznaczający jakie ryzyko popełnienia błędu I typu jesteśmy w stanie zaakceptować. Jak pisze Greń: *„do najczęściej przyjmowanych poziomów istotności należą prawdopodobieństwa* 0.1, 0.05, 0.01, 0.001*"*. Z tego powodu większość tablic statystycznych zawiera wartości progowe statystyk dla właśnie takich poziomów istotności. W przypadku ustalenia progu istotności wysokości powiedzmy 0.1, godzimy się na możliwość popełnienia 10 błędów na 100 podjętych decyzji odrzucenia hipotezy zerowej tj. co najwyżej 10 ze 100 odrzuconych hipotez zerowych będzie prawdziwych.

Jeśli nie narzucamy z góry poziomu istotności, wówczas możemy na podstawie wartości wyliczonej statystyki testowej wyznaczyć minimalne możliwe prawdopodobieństwo popełnienia błędu pierwszego rodzaju konieczne do przyjęcia hipotezy zerowej. Oczywistym jest, iż zastosowanie takiego podejścia implikuje uzyskanie dokładniejszej informacji na temat weryfikowanej hipotezy, bowiem ustalamy precyzyjniej jaką mamy szansę, że teza przyjętej hipotezy jest w rzeczywistości prawdziwa.

Ważnym pojęciem w teorii weryfikacji hipotez jest również błąd drugiego rodzaju nazywany także błędem typu II, błędem odrzucenia czy też beta-błędem. Polega on na przyjęciu hipotezy zerowej, która w rzeczywistości jest fałszywa. Za pomocą oszacowania prawdopodobieństwa tego błędu wyznacza się tak zwaną moc testu, czyli prawdopodobieństwo podjęcia

słusznej decyzji przy weryfikacji hipotezy, polegającej na odrzuceniu hipotezy fałszywej. Formalnie rzecz ujmując, jeśli przez $\beta^{(1)}$ oznaczymy prawdopodobieństwo popełnienia błędu typu II, to wówczas moc testu będzie równa $1 - \beta$.

Należy w tym miejscu podkreślić, iż w testach istotności weryfikowana hipoteza nigdy nie jest przyjmowana, nawet w przypadku stwierdzenia wysokiego prawdopodobieństwa co do jej prawdziwości. Podejmuje się jedynie decyzję odrzucenia sprawdzanej hipotezy lub stwierdza się, że nie ma podstaw do jej odrzucenia. Jest to uwarunkowane tym, iż w procedurze weryfikacyjnej uwzględnia się tylko błąd pierwszego rodzaju, pomijając szansę popełnienia błędu drugiego rodzaju. Krótko mówiąc testy istotności odrzucają hipotezy błędne, zaś o prawdziwych mówią jedynie tyle, że brak jest podstaw do ich odrzucenia. Aczkolwiek jak pisze Greń [1]:

> *„Istota rzeczy przy budowie każdego testu statystycznego polega jednak na tym, żeby uchronić się zarówno przed popełnieniem błędu pierwszego rodzaju, polegającym na odrzuceniu hipotezy prawdziwej, jak i przed popełnieniem błędu drugiego rodzaju, polegającym na przyjęciu hipotezy fałszywej."*

[1] Standardowo stosowane oznaczenie w literaturze.

Rozdział 3

Charakterystyka testów statystycznych

3.1. Różne kryteria podziału zbioru testów statystycznych

W tym podrozdziale przyjrzymy się trzem głównym kryteriom podziału zbioru testów statystycznych. W pierwszej kolejności zostanie opisany najpopularniejszy podział testów, mianowicie podział na testy parametryczne oraz nieparametryczne. Jak zobaczymy w literaturze panuje trochę zamieszania w definiowaniu własności tych dwóch klas testów. Następnie zostanie sprecyzowany podział testów ze względu na parowalność wyników prób. To kryterium jest w zasadzie najistotniejszym w badaniu istotności różnic algorytmów uczenia maszynowego, gdyż jest nieodzownie związane ze sposobem trenowania oraz testowania porównywanych metod. Ostatnim z przedstawionych kryteriów będzie kryterium podziału testów, ze względu na liczbę badanych prób. Wyróżnimy trzy główne grupy testów operujących kolejno na jednej, dwóch oraz większej liczbie próbek.

3.1.1. Testy parametryczne i nieparametryczne

Jest to jeden z najczęściej spotykanych, a zarazem najbardziej kontrowersyjny podział zbioru testów statystycznych jaki można znaleźć w literaturze. Kontrowersje są prawdopodobnie wynikiem małej istotności

znaczenia samych nazw tych dwóch grup testów, bowiem sama procedura przeprowadzania testu nie zależy od tego, do jakiej grupy on należy.

Pewne źródła [10] podają, iż testy parametryczne to takie testy, które zakładają rozkład normalny populacji generalnej. W innych [14] z kolei znajdujemy informację, że testy parametryczne to takie, w których czynimy jakiekolwiek założenie o rozkładzie obserwowanych danych. Faktem jest jednak, iż test McNemara wymagający danych o rozkładzie dwumianowym nie jest testem parametrycznym. Jeszcze inne źródła [15] twierdzą zaś, że są to testy, które *„służą weryfikacji hipotez parametrycznych, odnoszących się do parametrów rozkładu badanej cechy w populacji generalnej"*. Podobnego zdania jest Greń [1]. W literaturze można znaleźć również i takie poglądy autorów, które łączą powyższe definicje. Według nich testy parametryczne służą porównywaniu parametrów zmiennych losowych, przy czym wykonuje się je dla zmiennych o rozkładach nie odbiegających od normalnego.

Wydaje się, iż najbardziej naturalnym w tej sytuacji będzie przyjęcie definicji proponowanej przez Wikipedię [15] oraz Grenia [1]:

Def.: Testy parametryczne to testy służące weryfikacji hipotez parametrycznych, a więc takich, które odnoszą się do parametrów rozkładu badanej cechy w populacji generalnej (precyzują wartość parametru w ustalonym typie rozkładu).

Konsekwentnie testy nieparametryczne zdefiniujemy w następujący sposób:

Def.: Testy nieparametryczne to grupa testów statystycznych służących do weryfikacji hipotez nieparametrycznych, precyzujących ogólny typ (postać) rozkładu populacji generalnej.

Testy parametryczne są znacznie dokładniejszym narzędziem badawczym, ponieważ wykorzystują więcej informacji zawartych w próbie niż testy nieparametryczne. Jednakże są one bezradne w przypadku danych

jakościowych czy porządkowych, z którymi wyśmienicie radzą sobie te drugie. Ponadto testy parametryczne opisują własności jedynie jakiegoś zjawiska (parametru), często nie dając wystarczających podstaw do wysnuwania wniosków ogólnych.

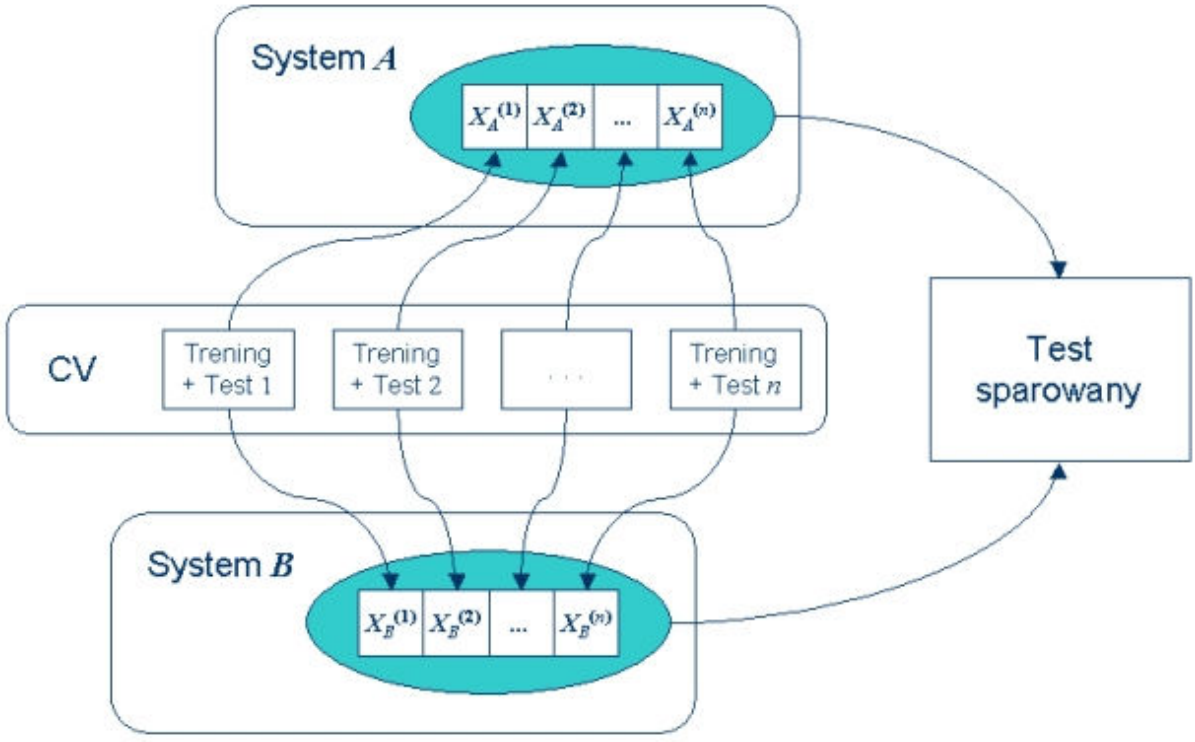
Najpopularniejszym przykładem testu parametrycznego jest test t-Studenta, w którym stawiana hipoteza zerowa, będąca właśnie hipotezą parametryczną, dotyczy średnich porównywanych próbek. Jest to jedyny test parametryczny omówiony w niniejszej pracy. Poza testami dla średnich wśród testów parametrycznych wyróżnia się także testy dla wariancji oraz testy dla wskaźnika struktury (procentów).

Grupa testów nieparametrycznych jest znacznie bogatsza. Testy te zastępują często testy parametryczne w przypadku, gdy nie są spełnione założenia stosowalności tych drugich. Najczęściej stosowanym w praktyce testem nieparametrycznym jest test χ^2. Prawdopodobnie wynika to z możliwości zastosowania go zarówno w badaniu cech mierzalnych jak i w szerokiej gamie problemów bazujących na danych jakościowych (skategoryzowanych), z którymi wiele testów sobie nie radzi. Kolejnym bardzo znanym i zarazem bardzo mocnym testem nieparametrycznym jest test kolejności par Wilcoxona dla prób sparowanych oraz jego wersja niesparowana test Manna-Whitneya-Wilcoxona. Jednakże w przypadku tych dwóch procedur weryfikacyjnych wymagane są dane porządkowe co nieco zawęża zakres adekwatnych sytuacji ich zastosowania. Innymi testami nieparametrycznymi jakie warto wymienić są: test Kołmogorowa-Smirnowa, test McNemara, test różnicy dwóch proporcji, test dokładności Fishera czy też test Andersona-Darlinga.

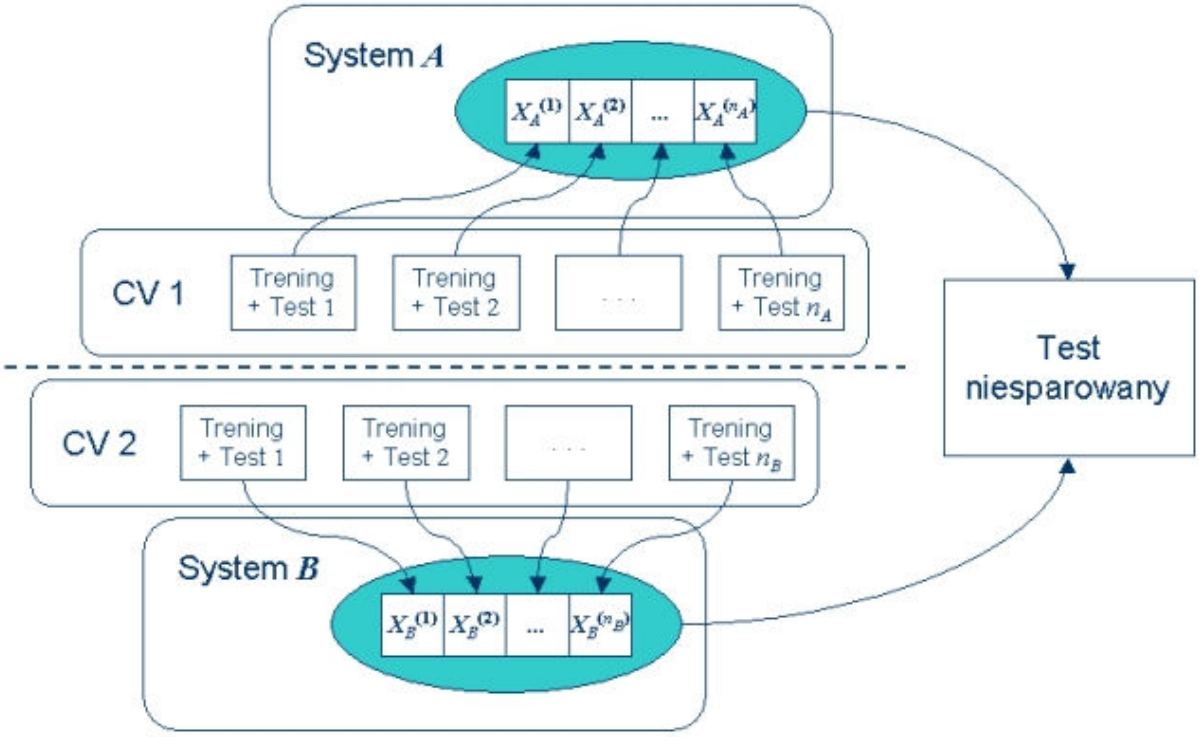
3.1.2. Testy sparowane a niesparowane

Kolejnym kryterium podziału zbioru testów statystycznych jest zależność pomiędzy próbkami. Jest to najważniejsze kryterium podziału ze względu na porównywanie metod uczenia maszynowego, gdyż warunkuje sposób ich trenowania oraz testowania.

Jeśli próbki są niezależne (nieskorelowane), to mówimy o teście niesparowanym, zaś gdy próbki są w pewien sposób ze sobą powiązane, to wówczas korzystamy z testu sparowanego. Różnica leży zatem w sposobie pobierania próbek z wynikami. W przypadku algorytmów uczenia maszynowego mamy dwie możliwości pobierania wyników (jak to zostało opisane w podrozdziale 2.1): pojedynczy trening + test lub wykorzystanie metod resamplingu danych. Różnicę pomiędzy testem sparowanym a niesparowanym łatwo przedstawić graficznie na przykładzie kroswalidacji:



Rysunek 3.1: Zbieranie wyników dla testu sparowanego.



Rysunek 3.2: Zbieranie wyników dla testu niesparowanego.

Jak widać w przypadku testu sparowanego oba algorytmy trenowane są na tych samych zbiorach danych. To samo tyczy się testowania utworzonych w fazie treningu klasyfikatorów. Tym samym poszczególne wartości uzyskanych wyników są ze sobą skorelowane. Ponadto łatwo jest wywnioskować, iż liczba uzyskanych w ten sposób wyników dla obu systemów będzie równa.

Przeciwnie sprawa ma się do testów niesparowanych. Algorytmy tworzą klasyfikatory zupełnie niezależnie, a następnie testują je na danych, które w żaden sposób nie muszą być ze sobą powiązane. Pociąga to za sobą brak korelacji pomiędzy uzyskiwanymi wynikami. Takie podejście pozwala na utworzenie wektorów wyników różnej długości.

Na koniec tego podrozdziału podsumowanie obu typów testów w zwięzłej formie tabeli:

Test sparowany	Test niesparowany
wektory z wynikami tylko tej samej wielkości	możliwa różna wielkość wektorów z wynikami
poszczególne wyniki powiązane ze sobą	brak powiązania pomiędzy poszczególnymi wynikami
wyniki modeli zbierane równolegle – na tych samych podzbiorach danych	wyniki modeli zbierane osobno – na różnych podzbiorach danych

Tabela 3.1: Podsumowanie testów sparowanych i niesparowanych.

3.1.3. Liczba porównywanych próbek

To kryterium podziału zbioru testów statystycznych dzieli go na trzy podzbiory:

- testy dla jednej próby,
- testy dla dwóch prób,
- testy dla większej liczby prób (więcej niż dwóch).

Do pierwszej grupy zalicza się testy sprawdzające, czy wyniki próby losowej pochodzą z populacji o określonym typie rozkładu lub co więcej z populacji o pewnych określonych parametrach.

Najczęściej stosowanym testem parametrycznym należącym do tej grupy jest test t-Studenta dla jednej próby. Bada on, czy posiadane wyniki pochodzą z populacji o określonej średniej. Gdy założenie o normalności rozkładu nie jest spełnione, można posłużyć się nieparametrycznym testem kolejności par Wilcoxona.

Z nieparametrycznych testów dla jednej próby ważną klasę stanowią tak zwane testy zgodności służące do weryfikacji hipotez, traktujących o rozkładzie zmiennej losowej. Porównują one rozkład danej zmiennej z pewnym

rozkładem teoretycznym. Najważniejszymi przedstawicielami testów zgodności są: test zgodności χ^2, test Kołmogorowa-Smirnowa, test Lillieforce'a, test Shapiro-Wilka oraz test Andersona-Darlinga. Testy te najczęściej stosuje się w celu sprawdzenia normalności rozkładu próby przed zastosowaniem testów parametrycznych, zakładających właśnie taki rozkład populacji generalnej.

Do drugiej grupy testów zaliczamy szeroko stosowane testy istotności dla prób zależnych oraz niezależnych. Służą one weryfikacji hipotez stwierdzających, że dwie próby pochodzą z jednej populacji lub inaczej mówiąc, że dwie populacje mają ten sam rozkład. Są to najczęściej wykorzystywane w praktyce testy statystyczne. W celu badania istotności różnic systemów uczących się posługujemy się testami właśnie z tej grupy.

Podstawowym testem parametrycznym dla dwóch zbiorów danych jest test t-Studenta zarówno w wersji dla prób sparowanych jak i niesparowanych. W teście tym weryfikujemy hipotezę o równości średnich badanych zbiorów w celu stwierdzenia, czy próby pochodzą z tej samej populacji. Jednym z ważniejszych testów parametrycznych wchodzących w skład tej grupy jest również test F na jednorodność wariancji.

Do nieparametrycznych testów operujących na dwóch próbach losowych należą takie testy jak test kolejności par Wilcoxona dla prób sparowanych i jego niesparowana odmiana test Manna-Whitneya-Wilcoxona, test McNemara, różnicy dwóch proporcji czy jeden z najczęściej stosowanych testów w praktyce test χ^2.

Trzecia grupa testów została podana jedynie w celu zobrazowania pełni zagadnienia tego kryterium podziału. Problematyka tych testów nie została poruszona z powodu małej przydatności w porównywaniu różnic metod uczenia maszynowego – nie ma potrzeby analizowania więcej niż dwóch algorytmów na raz. Te testy stosuje się często w zastosowaniach medycznych czy

chemicznych w celu porównywania wyników uzyskiwanych różnymi metodami badawczymi.

Zalicza się tutaj parametryczną analizę wariancji (w skrócie ANOVA) tak dla prób sparowanych jak i niesparowanych oraz jej nieparametryczne odmiany, mianowicie test Kruskala-Wallisa (wersja dla prób zależnych) i test Friedmana (dla prób niezależnych). Ponadto do porównywania jednorodności wariancji większej liczby prób można posłużyć się testem Bartletta, który jest rozszerzeniem testu F.

3.2. Testy istotności dla prób sparowanych

W tym podrozdziale zostaną szczegółowo omówione testy porównujące dwie zależne próby losowe. Dokładne różnice pomiędzy testami sparowanymi a niesparowanymi zostały omówione w podrozdziale 3.1.2, natomiast sposoby pobierania odpowiednich wektorów z wynikami w podrozdziale 2.1.

3.2.1. Test McNemara

Nazwa tego testu pochodzi od nazwiska Quinna McNemara, który zaproponował go w 1947 roku. Jest to nieparametryczny test dla dwóch skorelowanych zmiennych dychotomicznych, bazujący na teście χ^2 – porównujemy rozkład wartości oczekiwanych z rozkładem wartości zaobserwowanych. W literaturze spotykany jest również pod nazwą testu istotności zmian.

Dane:

- n – liczba wyników zero-jedynkowych (długość wektorów X_A i X_B) równa liczbie wektorów należących do zbioru testowego
- $X_A = [X_A^{(1)}, X_A^{(2)}, ..., X_A^{(n)}]^T$ – wektor z wynikami zero-jedynkowymi klasyfikatora f_A
- $X_B = [X_B^{(1)}, X_B^{(2)}, ..., X_B^{(n)}]^T$ – wektor z wynikami zero-jedynkowymi klasyfikatora f_B

Na podstawie próbek X_A oraz X_B tworzymy macierz kontyngencji zaobserwowanych wyników dwóch klasyfikatorów:

		f_A	
		błędnie sklas. (0)	poprawnie sklas. (1)
f_B	błędnie sklas. (0)	n_{00}	n_{10}
	poprawnie sklas. (1)	n_{01}	n_{11}

Tabela 3.2: Macierz kontyngencji wyników klasyfikatorów f_A i f_B.

Oczywiście wartości z tabeli 3.2 spełniają zależność:

$$n_{00} + n_{10} + n_{01} + n_{11} = n.$$

Zgodnie z hipotezą zerową (sformułowaną w podrozdziale 2.2) modele powinny robić tyle samo błędów:

$$n_{10} = n_{01}.$$

Zatem wartości oczekiwane przy prawdziwości hipotezy zerowej są następujące:

		f_A	
		błędnie sklas. (0)	poprawnie sklas. (1)
f_B	błędnie sklas. (0)	n_{00}	$\frac{n_{10}+n_{01}}{2}$
	poprawnie sklas. (1)	$\frac{n_{10}+n_{01}}{2}$	n_{11}

Tabela 3.3: Macierz kontyngencji wartości oczekiwanych klasyfikatorów f_A i f_B.

Rozbieżność pomiędzy zaobserwowanymi a oczekiwanymi wartościami mierzymy za pomocą statystyki zdefiniowanej w następujący sposób:

$$\chi^2 = \frac{(n_{01} - n_{10})^2}{n_{01} + n_{10}}.$$

Jej rozkład zbliżony jest do rozkładu χ^2 z 1 stopniem swobody.

Należy zauważyć, iż powyższa statystyka przyjmuje jedynie wartości dyskretne. Rozkład χ^2 jest natomiast rozkładem ciągłym. Z tego powodu wielu autorów stosuje korektę ciągłości w celu poprawienia przybliżenia zmiennej dyskretnej zmienną ciągłą. Statystyka uwzględniająca korektę ciągłości ma następującą postać:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}.$$

Według niektórych jednak taka technika jest kontrowersyjna.

Jak twierdzi Garson [9], przybliżenia rozkładu podanej statystyki do rozkładu χ^2 nie można stosować, jeśli n_{10} i n_{01} są małe ($n_{10} + n_{01} < 10$). Sugeruje on używanie testu dokładności Fishera w przypadku, gdy ten warunek nie jest spełniony. Lui [19] pokazał zaś eksperymentalnie, iż statystykę bez korekty ciągłości można stosować już dla $n_{10} + n_{01} \geq 6$.

Stwierdził również, że test bazujący na takiej statystyce ma większą moc od testu wykorzystującego statystykę z korektą ciągłości.

Największą wadą testu McNemara jest fakt, iż nie uwzględnia on wariancji spowodowanej wyborem zbioru treningowego oraz wewnętrznej wariancji algorytmów. Dlatego też, jak pisze Dietterich [2] *„testu McNemara powinno używać się w przypadku, gdy podejrzewamy, że te źródła zmienności są niewielkie"*.

3.2.2. Test różnicy dwóch proporcji

Jest to prosty test nieparametryczny mierzący różnicę pomiędzy błędami (prawdopodobieństwami popełnienia błędu) dwóch klasyfikatorów. Tak jak w przypadku testu McNemara wykorzystywane są próbki dwumianowe.

Dane:

- n – liczba wyników zero-jedynkowych (długość wektorów X_A i X_B) równa liczbie wektorów należących do zbioru testowego
- $X_A = [X_A^{(1)}, X_A^{(2)}, ..., X_A^{(n)}]^T$ – wektor z wynikami zero-jedynkowymi klasyfikatora f_A
- $X_B = [X_B^{(1)}, X_B^{(2)}, ..., X_B^{(n)}]^T$ – wektor z wynikami zero-jedynkowymi klasyfikatora f_B

W pierwszej kolejności wyliczamy prawdopodobieństwa popełnienia błędu odpowiednio przez klasyfikatory f_A oraz f_B. Skorzystamy tutaj z notacji wprowadzonej przy omawianiu testu McNemara (Tabela 3.2):

$$p_A = \frac{n_{00} + n_{01}}{n},$$

$$p_B = \frac{n_{00} + n_{10}}{n}.$$

Tak więc p_A (odpowiednio p_B) jest ilorazem liczby błędnie sklasyfikowanych wektorów przez klasyfikator f_A (odpowiednio f_B) do liczby wszystkich wektorów należących do zbioru testowego.

Rozważmy różnice:

$$d = p_A - p_B$$

jako jedną zmienną losową. Jak wiemy rozkład dwumianowy jest dobrze aproksymowany przez rozkład normalny dla odpowiednio dużych n. Dlatego możemy obydwa rozkłady dwumianowe przybliżyć przez rozkłady normalne (dla $n \geq 30$, $p_A \cdot n > 5$, $(1 - p_A) \cdot n > 5$, $p_B \cdot n > 5$, $(1 - p_B) \cdot n > 5$, Kuncheva [3]). Jeśli założymy, że oba prawdopodobieństwa popełnienia błędu są niezależne, to zmienna losowa d będzie posiadała rozkład normalny. Wynika to z faktu, iż różnica dwóch niezależnych ziemnych losowych posiadających rozkład normalny jest zmienną losową o rozkładzie normalnym.

Oznaczmy przez p średnie prawdopodobieństwo popełnienia błędu:

$$p = \frac{p_A + p_B}{2}.$$

Wówczas statystyka wyznaczona wzorem:

$$z = \frac{p_A - p_B}{\sqrt{\frac{2p(1-p)}{n}}}$$

ma w przybliżeniu rozkład normalny.

Zauważmy jednak, że prawdopodobieństwa p_A oraz p_B nie są niezależne. Jest to spowodowane tym, że do ich pomiaru używamy tego samego zbioru

testowego (klasyfikatory f_A oraz f_B są testowane na tym samym zbiorze), co implikuje brak niezależności. Pomimo takiego nadużycia powyższa statystyka jest szeroko stosowana w literaturze uczenia maszynowego. Dietterich [2] pokazał jednak eksperymentalnie jak bardzo negatywny skutek na dokładność testu ma błędne założenie o niezależności. Zaproponował on również poprawioną statystykę nie wymagającą owego założenia:

$$z' = \frac{|n_{01} - n_{10}| - 1}{\sqrt{n_{01} + n_{10}}}.$$

Łatwo zauważyć, iż tak zdefiniowana statystyka jest pierwiastkiem kwadratowym statystyki χ^2 z testu McNemara.

Pomimo wprowadzenia powyższej korekty poprawiającej dokładność testu Dietterich [2] proponuje jednak używanie testu McNemara zamiast testu różnicy dwóch proporcji.

Podobnie jak w przypadku testu McNemara dużym mankamentem jest fakt, iż test ten nie uwzględnia wariancji spowodowanej wyborem zbioru treningowego i testowego oraz wewnętrznej wariancji algorytmów.

3.2.3. Sparowany test t-Studenta

Nazwa tego testu pochodzi od pseudonimu literackiego – „Student" – jego twórcy Williama Sealy'ego Gosseta, który opublikował go w 1908 roku. Jest to jeden z najpopularniejszych testów spotykanych w literaturze (również literaturze uczenia maszynowego). Fakt ten wynika z prostoty jego budowy, łatwości w użyciu oraz możliwości zastosowania w szerokiej gamie sytuacji. W tym podrozdziale omówimy wersję sparowaną tego testu, zaś w podrozdziale 3.3.3 przyjrzymy się jego niesparowanej wersji.

Sparowany test t-Studenta jest parametrycznym testem mierzącym istotność różnicy pomiędzy średnimi dwóch skorelowanych próbek. Dokładniej mówiąc bada, czy średnia różnic pomiędzy parami pomiarów istotnie różni się od zera. Jak podaje IFA [6] test ten jest najdokładniejszym testem dla danych interwałowych, jednakże posiada również najbardziej rygorystyczne założenia.

Dane:

- n – liczba zebranych wyników algorytmów (długość wektorów X_A i X_B)
- $X_A = [X_A^{(1)}, X_A^{(2)}, ..., X_A^{(n)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu A
- $X_B = [X_B^{(1)}, X_B^{(2)}, ..., X_B^{(n)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu B

Na podstawie wektorów X_A oraz X_B wyznaczamy wektor różnic poszczególnych par wyników:

$$D^{(i)} = X_A{}^{(i)} - X_B{}^{(i)}.$$

W następnej kolejności wyliczamy średnią wektora D:

$$\mu_D = \sum_{i=1}^{n} \frac{D^{(i)}}{n}$$

oraz jego standardowe odchylenie:

$$\sigma_D = \sqrt{\frac{\sum_{i=1}^{n} (D^{(i)} - \mu_D)^2}{n-1}}.$$

Statystyka testowa zadana jest wzorem:

$$t = \frac{\mu_D}{\sigma_D}\sqrt{n}.$$

Posiada ona rozkład t-Studenta z $n - 1$ stopniami swobody.

Powyższą statystykę możemy przekształcić do prostszej obliczeniowo postaci (wyciągamy tylko jeden pierwiastek zamiast dwóch):

$$t = \frac{\mu_D}{\sigma_D}\sqrt{n} = \frac{\mu_D}{\sqrt{\frac{\sum_{i=1}^{n}(D^{(i)} - \mu_D)^2}{n(n-1)}}}.$$

Jest to przydatne przy implementacji tego testu.

IFA [6] podaje, iż w przypadku, gdy liczba stopni swobody jest większa od 30, to rozkład t może być przybliżony przez rozkład normalny.

Lowry [4] wymienia trzy zasadnicze założenia stosowalności sparowanego testu t-Studenta:

- wartości różnic par pomiarów są losowo pobrane z populacji źródłowej;
- mamy podstawy, żeby przypuszczać, iż populacja generalna, z której pochodzą wartości różnic, posiada rozkład normalny;
- skala pomiarów dla obu próbek posiada własności skali równego interwału.

Założenie o normalności rozkładu możemy sprawdzić za pomocą popularnych testów zgodności takich jak test Kołmogorowa-Smirnowa, test Shapiro-Wilka czy test Andersona-Darlinga. W przypadku, gdy założenie to nie jest spełnione do celu badania istotności różnicy algorytmów możemy posłużyć się testem kolejności par Wilcoxona.

3.2.4. Test kolejności par Wilcoxona

W literaturze test ten spotykany jest również pod nazwą testu rangowanych znaków Wilcoxona. Jego nazwa pochodzi od nazwiska Franka Wilcoxona, który zaproponował go w 1945 roku. Jest on nieparametryczną alternatywą testu t-Studenta dla dwóch skorelowanych próbek. IFA [6] podaje, że dla dużych liczb test kolejności par Wilcoxona jest prawie tak dokładny jak test t-Studenta i sugeruje jego użycie (przynajmniej dla małych prób tj. gdy $n < 50$) zamiast testu t-Studenta z powodu dużej wrażliwości tego drugiego na odchylenia od rozkładu normalnego.

Podobnie jak w sparowanym teście t-Studenta procedura weryfikacyjna opiera się o porównywanie różnic pomiędzy pomiarami. Dlatego warunkiem koniecznym jest aby skala pomiarów miała własności skali porządkowej. Brak jest jednak założenia o rozkładzie populacji źródłowej, które jest największym ograniczeniem testu t-Studenta.

Dane:

- n – liczba zebranych wyników algorytmów (długość wektorów X_A i X_B)
- $X_A = [X_A^{(1)}, X_A^{(2)}, ..., X_A^{(n)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu A
- $X_B = [X_B^{(1)}, X_B^{(2)}, ..., X_B^{(n)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu B

W literaturze można spotkać pewne rozbieżności jeśli chodzi o algorytm wyliczania statystyki tego testu. Są one jednak na tyle nieznaczne, że nie ma potrzeby ich omawiania.

Tak jak w przypadku sparowanego testu t-Studenta zaczynamy od wyznaczenia wektora różnic poszczególnych par pomiarów:

$$D^{(i)} = X_A{}^{(i)} - X_B{}^{(i)}.$$

W następnej kolejności porządkujemy wartości bezwzględne różnic $D^{(i)}$ w sposób rosnący, po czym przypisujemy im rangi. W przypadku remisów (równych wartości $D^{(i)}$) nadajemy im tak zwane rangi wiązane, czyli średnią arytmetyczną z rang, jakie powinno się im przypisać, gdyby ich wartości nie były równe. Podczas tego procesu pomijamy różnice zerowe, ponieważ nie niosą one żadnej użytecznej informacji. Oznaczmy przez N liczbę istotnych różnic w rankingu, czyli wszystkich różnic oprócz pominiętych.

W kolejnym kroku sumujemy osobno rangi różnic dodatnich (oznaczymy je przez W_+) i ujemnych (odpowiednio przez W_-) oraz wyznaczamy maksimum z tych dwóch wartości:

$$W = \max(W_+, W_-).^{(2)}$$

W przypadku, gdy liczba N istotnych różnic w rankingu jest większa od 15, następująca statystyka ma w przybliżeniu rozkład normalny:

$$z = \frac{W - \frac{1}{4}N(N+1) - 0.5}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}}.$$

Dla $N \leq 20$ można wyliczać dokładne prawdopodobieństwo. Wówczas za statystykę testową przyjmuje się minimum z W_+ i W_-.

Lowry [4] wymienia następujące trzy założenia stosowalności tego testu:

- wartości par wyników są niezależnie i losowo pobrane z populacji źródłowej (każda para jest niezależnie pobierana od każdej innej pary);
- zmienna zależna jest ciągła (zdolna do wyprodukowania pomiarów z dokładnością do n-tego miejsca dziesiętnego);

[2] W literaturze statystka W oznaczana jest również literą T oraz S.

- skala pomiarów dla obu próbek posiada własności przynajmniej skali porządkowej, co jest potrzebne do utworzenia rankingu różnic.

3.2.5. Test 5x2cv Diettericha

Jest to charakterystyczny test, będący modyfikacją testu t-Studenta dla prób zależnych, zaproponowany przez Diettericha [2] w 1998 roku. Wykorzystuje się w nim specyficzny sposób pobierania wyników porównywanych algorytmów, skąd też wywodzi się jego nazwa 5x2cv (pięć powtórzeń dwukrotnej kroswalidacji).

Dietterich [2] zauważył, że w przypadku sparowanego testu t-Studenta, z powodu korelacji pomiędzy poszczególnymi wynikami uzyskanymi w procesie kroswalidacji statystyka wyliczana w tym teście miała często bardzo wysoką wartość. Jak twierdzi, dzieje się tak z powodu zbyt niskiego oszacowania wariancji (znajdującej się w mianowniku statystyki t) przy jednoczesnym słabym oszacowaniu średniej (w liczniku) z wyników próby. Spostrzegł on, że jeśli zamienimy licznik statystyki t (średnią) na zaobserwowaną różnicę z jednego przebiegu k-krotnej kroswalidacji, to wówczas statystyka ta staje się „spokojniejsza".

W teście tym wykonujemy pięć powtórzeń dwukrotnej kroswalidacji. Każde takie powtórzenie generuje 4 błędy klasyfikacji testowanych systemów A i B: $p_A^{(1)}$ i $p_B^{(1)}$ (otrzymane w pierwszej z dwóch iteracji algorytmu kroswalidacji) oraz $p_A^{(2)}$ i $p_B^{(2)}$ (otrzymane w drugiej iteracji).

Oznaczmy w następujący sposób różnice pomiędzy powyższymi błędami klasyfikacji:

$$p^{(1)} = p_A^{(1)} - p_B^{(1)},$$

$$p^{(2)} = p_A^{(2)} - p_B^{(2)}.$$

Średnia oraz wariancja dwóch takich różnic wynoszą odpowiednio:

$$\overline{p} = \frac{p^{(1)} + p^{(2)}}{2},$$

$$s^2 = \left(p^{(1)} - \overline{p}\right)^2 + \left(p^{(2)} - \overline{p}\right)^2.$$

Jeśli przez s_i^2, i = 1, ..., 5, oznaczymy wariancję wyliczoną na podstawie różnic błędów otrzymanych w i-tym powtórzeniu kroswalidacji, to wówczas statystyka testowa będzie miała postać:

$$\tilde{t} = \frac{p_1^{(1)}}{\sqrt{\frac{1}{5}\sum_{i=1}^{5} s_i^2}},$$

przy czym $p_1^{(1)}$ w liczniku oznacza różnicę $p^{(1)}$ z pierwszej kroswalidacji. Dietterich [2] udowodnił, iż tak zdefiniowana statystyka ma w przybliżeniu rozkład t-Studenta z 5 stopniami swobody.

3.3. Testy istotności dla prób niesparowanych

W tym podrozdziale zostaną przedstawione niesparowane testy istotności mające zastosowanie w porównywaniu wyników uzyskiwanych przez algorytmy uczenia maszynowego. Różnice pomiędzy testami dla prób skorelowanych i nieskorelowanych zostały omówione w podrozdziale 3.1.2.

3.3.1. Test χ^2

Test χ^2 jest jednym z najpopularniejszych nieparametrycznych testów istotności stosowanych w praktyce. Wykorzystywany jest do badania zarówno

cech mierzalnych jak i niemierzalnych (skategoryzowanych). Porównujemy w nim wartości empiryczne (zaobserwowane) z wartościami oczekiwanymi. Przy pomocy tego testu weryfikuje się jedynie hipotezy nieukierunkowane, co wynika ze sposobu jego budowy.

Dane:

- n_A – liczba wyników zero-jedynkowych klasyfikatora f_A (długość wektora X_A) równa liczbie wektorów należących do zbioru testowego tego klasyfikatora
- n_B – liczba wyników zero-jedynkowych klasyfikatora f_B (długość wektora X_B) równa liczbie wektorów należących do zbioru testowego tego klasyfikatora
- $X_A = [X_A^{(1)}, X_A^{(2)}, ..., X_A^{(n_A)}]^T$ – wektor z wynikami zero-jedynkowymi klasyfikatora f_A
- $X_B = [X_B^{(1)}, X_B^{(2)}, ..., X_B^{(n_B)}]^T$ – wektor z wynikami zero-jedynkowymi klasyfikatora f_B

Tworzymy tabelę dwudzielną, zawierającą wyniki klasyfikacji wektorów ze zbioru testowego uzyskane przez klasyfikatory f_A oraz f_B:

	liczba sklasyfikowanych wektorów		
	błędnych (0)	poprawnych (1)	ogółem
klasyfikator f_A	n_{A0}	n_{A1}	n_A
klasyfikator f_B	n_{B0}	n_{B1}	n_B
ogółem	N_0	N_1	N

Tabela 3.4: Tabela dwudzielna wyników klasyfikatorów f_A oraz f_B.

Wartości w powyższej tabeli opisane są następującymi zależności:

$$n_A = n_{A0} + n_{A1},$$
$$n_B = n_{B0} + n_{B1},$$
$$N_0 = n_{A0} + n_{B0},$$
$$N_1 = n_{A1} + n_{B1},$$
$$N = n_A + n_B = N_0 + N_1.$$

Tabela dwudzielna wartości oczekiwanych będzie miała następującą postać:

	liczba sklasyfikowanych wektorów		
	błędnych (0)	poprawnych (1)	ogółem
klasyfikator f_A	$\frac{n_A N_0}{N}$	$\frac{n_A N_1}{N}$	n_A
klasyfikator f_B	$\frac{n_B N_0}{N}$	$\frac{n_B N_1}{N}$	n_B
ogółem	N_0	N_1	N

Tabela 3.5: Tabela wartości oczekiwanych dla klasyfikatorów f_A oraz f_B.

Podstawowa statystyka testowa χ^2, zwana statystyką χ^2 Pearsona zadana jest wzorem:

$$\chi^2 = \sum_{i=1}^{4} \frac{(O_i - E_i)^2}{E_i},$$

przy czym O_i, i = 1, ..., 4, oznaczają zaobserwowane wartości zmiennych losowych, czyli w naszym przypadku wartości n_{A0}, n_{A1}, n_{B0}, n_{B1} z powyższej Tabeli 3.4, zaś E_i, i = 1, ..., 4, odpowiadające im wartości oczekiwane (Tabela 3.5).

Jak podaje Lowry [4], jeśli w tabeli wielodzielnej są dokładnie dwa wiersze i dwie kolumny, to statystyka χ^2 Pearsona wymaga korekty ciągłości. Polega ona na zmniejszeniu bezwzględnej wartości różnic między liczebnościami empirycznymi a oczekiwanymi o wartość 0.5 przed podniesieniem do kwadratu. Tak zdefiniowaną korektę nazywa się korektą ciągłości Yatesa od nazwiska jej twórcy Franka Yatesa, który zaproponował ją w 1934 roku. Niektóre źródła [12] podają, że tą poprawkę należy stosować dopiero w przypadku, gdy pewne wartości oczekiwane (Tabela 3.5) są mniejsze od 10.

Poprawiona statystyka z korektą ciągłości Yatesa ma następującą postać:

$$\chi^2 = \sum_{i=1}^{4} \frac{\left(|O_i - E_i| - 0.5\right)^2}{E_i}.$$

Test χ^2 wykorzystujący tak poprawioną statystykę nazywa się często testem χ^2 Yatesa.

Zarówno pierwsza jak i druga z podanych statystyk posiada rozkład χ^2 z jednym stopniem swobody.

Lowry [4] wymienia następujące dwa założenia jakie muszą być spełnione w celu przeprowadzenia tego testu:

- kategorie, na które podzielone są obserwacje są niezależne (wykluczają się wzajemnie);
- wszystkie liczebności oczekiwane są większe lub równe 5 (Greń [1] podaje wartość 8).

W przypadku, gdy niespełnione jest założenie dotyczące wielkości wartości oczekiwanych, to wówczas możemy posłużyć się testem dokładności Fishera.

3.2.2 Test F na jednorodność wariancji

Test ten często nazywany jest testem F-Snedecora, jako że został zaproponowany przez George'a Waddella Snedecora w 1934 roku. Literę F Snedecor nadał mu na cześć Ronalda Aylmera Fishera, który jest pierwotnym twórcą analizy wariancji. W literaturze test ten można również spotkać pod nazwą testu Fishera, czy testu Fishera-Snedecora.

Test F jest prostym parametrycznym testem statystycznym stosowanym w celu sprawdzenia istotności różnicy w stopniu rozproszenia badanej cechy w populacjach, z których pochodzą wyniki dwóch niezależnych prób. Jest on często wykorzystywany do weryfikacji spełnialności założenia o jednorodności wariancji przed wykonaniem testu t-Studenta.

Dane:

- n_A – liczba zebranych wyników algorytmu A (długość wektora X_A)
- n_B – liczba zebranych wyników algorytmu B (długość wektora X_B)
- $X_A = [X_A^{(1)}, X_A^{(2)}, ..., X_A^{(n_A)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu A
- $X_B = [X_B^{(1)}, X_B^{(2)}, ..., X_B^{(n_B)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu B

Do wyznaczenia statystyki testowej potrzebujemy oszacowania wariancji obu populacji. Greń [1] zaleca stosowanie do tego celu estymatorów asymptotycznie nieobciążonych (zwanych obciążonymi), które zadane są wzorami:

$$s_{X_A}^2 = \frac{1}{n_A - 1} \sum_{i=1}^{n_A} (X_A^{(i)} - \mu_{X_A})^2 ,$$

$$s_{X_B}^2 = \frac{1}{n_B - 1} \sum_{i=1}^{n_B} (X_B^{(i)} - \mu_{X_B})^2 \ .$$

Na podstawie oszacowań wariancji populacji generalnych wyznaczamy statystykę testową w następujący sposób:

$$F = \frac{s_{X_A}^2}{s_{X_B}^2}.$$

Przy prawdziwości hipotezy zerowej ma ona rozkład F-Snedecora. W przypadku tej statystyki mamy do czynienia z dwoma rodzajami stopni swobody: licznika i mianownika. Dla naszych danych będą to odpowiednio liczby (n_A – 1) oraz (n_B – 1).

Test F zakłada, iż rozkład badanej cechy w populacji generalnej jest normalny, dlatego przed jego wykonaniem powinno dokonać się sprawdzenia normalności rozkładu obu prób za pomocą jednego z testów zgodności.

3.3.3. Niesparowany test t-Studenta

W podrozdziale 3.2.3 została zaprezentowana sparowana wersja tego testu. W wersji dla prób niezależnych podstawową różnicę stanowi wyliczana w nim statystyka testowa. Pomimo tej rozbieżności jest on bardzo podobny do wersji dla prób skorelowanych.

Niesparowany test t-Studenta jest testem parametrycznym mierzącym istotność różnicy pomiędzy średnimi dwóch niezależnych próbek. Dokładniej mówiąc bada, czy różnica pomiędzy średnimi istotnie różni się od zera.

Dane:

- n_A – liczba zebranych wyników algorytmu A (długość wektora X_A)

- n_B – liczba zebranych wyników algorytmu B (długość wektora X_B)
- $X_A = [X_A^{(1)}, X_A^{(2)}, ..., X_A^{(n_A)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu A
- $X_B = [X_B^{(1)}, X_B^{(2)}, ..., X_B^{(n_B)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu B

Wyliczamy wariancję sumaryczną próbek, będącą oszacowaniem wariancji populacji generalnej:

$$s^2 = \frac{\sum_{i=1}^{n_A}(X_A^{(i)} - \mu_{X_A})^2 + \sum_{i=1}^{n_B}(X_B^{(i)} - \mu_{X_B})^2}{(n_A - 1) + (n_B - 1)}.$$

Przy pomocy powyższego oszacowania oraz wyliczonych średnich z próbek X_A i X_B budujemy statystykę testową postaci:

$$t = \frac{\mu_{X_A} - \mu_{X_B}}{\sqrt{s^2\left(\frac{1}{n_A} + \frac{1}{n_B}\right)}}.$$

Posiada ona rozkład t-Studenta z ($n_A + n_B - 2$) stopniami swobody.

Jak podaje IFA [6], jeśli liczba stopni swobody jest większa od 30, to rozkład t może być przybliżony przez rozkład normalny.

Lowry [4] wymienia następujące trzy założenia stosowalności niesparowanego testu t-Studenta:

- próbki są niezależnie i losowo pobrane z populacji źródłowej;
- mamy podstawy, żeby przypuszczać, iż populacja źródłowa posiada rozkład normalny;
- skala pomiarów dla obu próbek posiada własności skali równego interwału.

W przypadku, gdy naruszone jest założenie o normalności rozkładu pobranych próbek w miejsce tego testu stosuje się nieparametryczny test Manna-Whitneya-Wilcoxona. Tak jak w sparowanym teście t-Studenta do celu sprawdzenia normalności rozkładów możemy posłużyć się jednym z testów zgodności.

Ponadto w wielu źródłach [1, 6, 12] można znaleźć informację, iż test ten wymaga spełnienia dodatkowego założenia. Chodzi mianowicie o równość wariancji populacji generalnych, z których pochodzą wyniki porównywanych prób. Do sprawdzenia tego warunku najczęściej wykorzystuje się parametryczny test F-Snedecora.

Niestety w praktyce porównywania wyników algorytmów uczenia maszynowego sytuacja równych wariancji należy raczej do rzadkości. Jeśli wariancje obu prób nie są równe, wówczas do badania istotności różnicy możemy wykorzystać następującą statystykę testową:

$$t = \frac{\mu_{X_A} - \mu_{X_B}}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}},$$

przy czym s_A^2 i s_B^2 są wariancjami wektorów odpowiednio X_A oraz X_B.

3.3.4. Test Manna-Whitneya-Wilcoxona

Test ten zaproponowany został niezależnie przez Manna i Whitneya w 1947 roku oraz Wilcoxona w 1945. Z tego powodu w literaturze występuje pod różnymi nazwami. I tak oprócz nazwy stosowanej przez autora niniejszej pracy określany jest również mianem testu sumy rang Wilcoxona oraz testu U Manna-Whitneya (U ze względu na standardowo stosowane oznaczenie wyliczanej w nim statystyki).

Test Manna-Whitneya-Wilcoxona jest jednym z najmocniejszych testów nieparametrycznych. Stanowi on alternatywę dla niesparowanego testu t-Studenta, którego założenie o normalności rozkładu populacji generalnej może być często dyskwalifikujące. Test ten opiera się o miarę istotności różnicy pomiędzy medianami dwóch niezależnych próbek. Zalicza się go do grupy tak zwanych testów permutacyjnych lub ogólniej testów kombinatorycznych.

Dane:

- n_A – liczba zebranych wyników algorytmu A (długość wektora X_A)
- n_B – liczba zebranych wyników algorytmu B (długość wektora X_B)
- $X_A = [X_A^{(1)}, X_A^{(2)}, ..., X_A^{(n_A)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu A
- $X_B = [X_B^{(1)}, X_B^{(2)}, ..., X_B^{(n_B)}]^T$ – wektor z wynikami uzyskanymi podczas testowania algorytmu B

Algorytm wyznaczania statystyki rozpoczynamy od posortowania w sposób rosnący wartości obu próbek łącznie. Następnie tak uporządkowanym wartościom przypisujemy rangi. Remisy, jak w przypadku testu kolejności par Wilcoxona, dostają tak zwane rangi wiązane, będące średnią arytmetyczną z rang, jakie powinno się im przypisać, gdyby ich wartości nie były równe.

Oznaczmy przez W_A sumę rang przypisanych wartościom z próbki X_A oraz przez W_B sumę rang przypisanych wartościom z próbki X_B.

Lowry [4] podaje dwie metody dalszego postępowania – obie obarczone pewnymi wadami.

Metoda I

Pierwsza metoda bazuje na dodatkowym założeniu określającym minimalną długość próbek X_A oraz X_B. Wykorzystuje ona fakt, iż o ile n_A i n_B są odpowiednio duże, to wówczas W_A oraz W_B mają w przybliżeniu rozkład normalny. W literaturze różni autorzy podają różne wartości progowe dla n_A i n_B, od których można stosować owo przybliżenie do rozkładu normalnego. I tak u Lowry'a [4] znajdziemy wartość 5. Z kolei Dreyfus i Guyon [8] podają wartość 7. Najwyższy próg zaś, bo aż 10, stawia IFA [6].

Załóżmy zatem, że wektory z wynikami mają długości odpowiednio duże, to znaczy takie, które pozwalają zastosować wspomniane powyższej przybliżenie, powiedzmy n_A oraz n_B większe od 5.

Wyliczamy wartości oczekiwane statystyk W_A i W_B, czyli średnie sumy rang dla próbek wielkości odpowiednio n_A i n_B:

$$E(W_A) = \mu_{W_A} = n_A \frac{(n_A + n_B + 1)}{2},$$

$$E(W_B) = \mu_{W_B} = n_B \frac{(n_A + n_B + 1)}{2},$$

a następnie ich odchylenie standardowe (równe dla W_A i W_B):

$$\sigma = \sqrt{\frac{n_A n_B (n_A + n_B + 1)}{12}}.$$

Za pomocą wyznaczonych wartości wyliczamy końcowe statystyki testowe, które w tym przypadku mają postać:

$$z_A = \frac{(W_A - \mu_A) \pm 0.5}{\sigma},$$

$$z_B = \frac{(W_B - \mu_B) \pm 0.5}{\sigma}.$$

W powyższych wzorach stosujemy następujące korekty ciągłości (oznaczone przez ±0.5) w celu uwzględnienia faktu, iż rozkład W_A i W_B jest (z natury) dyskretny:

- -0.5, gdy $W_A > \mu_A$ (odpowiednio $W_B > \mu_B$),
- $+0.5$, gdy $W_A < \mu_A$ (odpowiednio $W_B < \mu_B$).

W ten sposób zdefiniowane statystyki mają w przybliżeniu rozkład normalny. Oczywiście nie ma potrzeby wyliczania ich obu. Wystarczy wykorzystać tylko jedną z nich, gdyż są one równe co do wartości bezwzględnej:

$$|z_A| = |z_B| .$$

Dreyfus i Guyon [8] używają zawsze pierwszej próbki do wyliczenia statystyki. W praktyce programistycznej lepiej jednak do tego celu posłużyć się próbką o mniejszej liczbie elementów, gdyż w ten sposób zmniejsza się liczba wykonywanych obliczeń.

Metoda II

Druga metoda nie wymaga dodatkowego założenia (co do wielkości próbek) jednakże jest obliczeniowo bardzo złożona. O ile wyliczenie statystyki nie stanowi żadnego problemu, o tyle wyznaczenie dokładnego progu istotności, jest nietrywialnym zadaniem ponieważ opiera się o wyznaczenie wszystkich możliwych permutacji rang pomiędzy dwiema próbkami (o długościach n_A i n_B). Jak twierdzi IFA [6], problem ten staje się obliczeniowo trudny dla n_A oraz n_B większych od 7. Liczba wszystkich możliwych rozmieszczeń ($n_A + n_B$) rang w dwóch grupach o wielkościach n_A i n_B jest równa:

$$\frac{(n_A + n_B)!}{n_A! n_B!} .$$

Dla $n_A = n_B = 7$ wszystkich rozmieszczeń jest 3 432, zaś dla $n_A = n_B = 10$ jest już ich 184 756.

Statystykę stosowaną w tej metodzie standardowo oznacza się literą U. Jej wartość jest równa różnicy pomiędzy maksymalną możliwą wartością W dla danej wielkości próbki (oznaczymy ją odpowiednio przez $W_{A[\max]}$ oraz $W_{B[\max]}$) a wartością W wyliczoną za pomocą danych wejściowych:

$$U_A = W_{A[\max]} - W_A,$$
$$U_B = W_{B[\max]} - W_B.$$

Oczywiście wartości $W_{A[\max]}$ oraz $W_{B[\max]}$ wyznaczamy według wzorów:

$$W_{A[\max]} = \sum_{i=n_B+1}^{n_A+n_B} i = n_A n_B + \frac{n_A(n_A+1)}{2},$$
$$W_{B[\max]} = \sum_{i=n_A+1}^{n_A+n_B} i = n_A n_B + \frac{n_B(n_B+1)}{2}.$$

Tak jak w przypadku pierwszej metody nie ma potrzeby obliczania obu statystyk. Ich wartości są bowiem symetrycznie odbite względem środka rozkładu, jakim jest wartość oczekiwana statystyk U_A i U_B, czyli wartość, której spodziewamy się przy prawdziwości hipotezy zerowej. Jest ona zadana wzorem:

$$E(U_A) = E(U_B) = \frac{n_A n_B}{2}.$$

Jak podaje Lowry [4] test Manna-Whitneya-Wilcoxona wymaga spełnienia następujących warunków:

- wyniki są niezależnie i losowo pobrane z populacji źródłowej
- zmienna zależna jest ciągła (pomiary z dokładnością do n-tego miejsca dziesiętnego)

- skala pomiarów dla obu próbek posiada własności przynajmniej skali porządkowej (co jest potrzebne do utworzenia rankingu wartości)

Ponadto w niektórych źródłach [18] można znaleźć informację, iż tego testu nie powinno stosować się w przypadku, gdy remisy (równe wartości wyników) stanowią dużą część wszystkich pomiarów.

3.4. Podsumowujące zestawienie testów

Nazwa testu	Skala pomiarowa	Założenie o normalności	Wielkość prób[3]	Rozkład statystyki	Inne założenia
Testy sparowane					
McNemara	dwumianowa	nie	> 30 [4]	χ^2	$n_{01} + n_{10} \geq 10$
różnicy dwóch proporcji	dwumianowa	nie	-	normalny	-
sparowany t-Studenta	równego interwału	tak	≤ 30	t-Studenta	-
			> 30	normalny	
kolejności par Wilcoxona	porządkowa	nie	> 15	normalny	ciągła zmienna zależna
			≤ 20	W	
5x2cv Dietterichа	równego interwału	tak	10	t-Studenta	-
Testy niesparowane					
χ^2	dowolna	nie	> 40 [5]	χ^2	$\forall_i E_i \geq 5$ [6]
F	równego interwału	tak	-	F-Snedecora	-
niesparowany t-Studenta	równego interwału	tak	≤ 30	t-Studenta	równe wariancje obu prób
			> 30	normalny	
Manna-Whitneya-Wilcoxona	porządkowa	nie	> 5	normalny	ciągła zmienna zależna
			-	U	

Tabela 3.6: Podsumowanie testów istotności omówionych w pracy.

[3] Przykładowe wartości progowe jakie można spotkać w literaturze. Różne źródła podają jednak różne wartości, od których można stosować dany test, dlatego powinno traktować się je z dużym dystansem.
[4] W literaturze można spotkać również wartość 50.
[5] Dotyczy sumy długości dwóch prób. Niektóre źródła podają wartość 20, inne aż 50.
[6] Wszystkie wartości oczekiwane (E_i) są większe lub równe 5. Szczegóły w rozdziale 3.3.1.

Rozdział 4

Opis pluginu testów statystycznych w systemie Intemi

4.1. Ogólna architektura Intemi i jej konsekwencje dla pluginu

System Intemi (INTElligent MIning) jest powstającym obecnie nowym systemem służącym do inteligentnej analizy danych, czyli tak zwanego „data miningu". Jego głównym przeznaczeniem jest wykorzystanie algorytmów uczenia maszynowego do badania różnego rodzaju danych. Znajdziemy w nim na przykład szeroko rozpowszechnione w ostatnich czasach sieci neuronowe, proste ale i skuteczne drzewa decyzyjne, różnego rodzaju wizualizatory oraz metody selekcji cech jak i całą gamę innych algorytmów Inteligencji Obliczeniowej. Ponadto dane poddawane badaniu w systemie Intemi mogą być preparowane na przeróżne sposoby w zależności od potrzeb konkretnej analizy.

4.1.1. Ogólna architektura Intemi

Od strony programistycznej niewątpliwie największą zaletą systemu Intemi jest jego modularność. Cały program dzieli się na uniwersalne jądro

oraz niezależne modele realizujące określone zadania. Z powodu takiego podziału systemu uzyskujemy możliwość prostego rozbudowywania środowiska o nowe modele realizujące nowe funkcje. Oczywiście za taką modularnością stoją pewne ograniczenia (warunki) dotyczące tworzenia samych modeli.

Ogólny schemat życia każdego modelu w systemie Intemi, czyli klasy implementującej interfejs `IModel`, składa się z czterech etapów: konfiguracja, tworzenie, uruchamianie a następnie korzystanie z niego.

Konfiguracja modelu odbywa się przy wykorzystaniu pewnej klasy konfiguracyjnej (wskazanej atrybutem `ConfigAttribute` w klasie modelu) implementującej interfejs `IConfiguration`. Za pomocą takiej klasy ustawiane są parametry oraz definiowane wejścia i wyjścia modelu. Zarówno wejścia jak i wyjścia definiuje się przy użyciu pewnych interfejsów. Są to zatem jakieś obiekty, będące instancjami klas implementującymi określone interfejsy.

Model uzyskuje dostęp do wszystkich parametrów oraz wejść przy wykorzystaniu metod interfejsu `IModelBase`. Obiekt takiego typu ustawia w modelu jądro systemu korzystając z metody `SetModelBase()` interfejsu `IModel`. Parametry modelu uzyskujemy za pomocą odpowiednich właściwości, zaś wejścia otwierane są przy użyciu metody `OpenInput()`.

Interfejs `IModel` wymaga zaimplementowania w klasie modelu metody `Run()`, która wywoływana jest przez jądro systemu. Metoda ta ma służyć do uruchomienia modelu, a co za tym idzie utworzenia odpowiedniego, na ogół w pełni zamkniętego obiektu, który następnie wystawiany jest jako wyjście modelu. Oczywiście klasa takiego obiektu musi implementować interfejs zdefiniowany podczas konfiguracji wyjścia modelu. Ponadto każdy interfejs definiujący wyjście musi rozszerzać interfejs `IOutput`, będący bazowym interfejsem dla wszystkich wyjść.

Obiekty wystawiane na wyjściu modeli udostępniają metody dostarczające pewną funkcjonalność. Przykładowo obiekt typu `IClassifier` (reprezentujący klasyfikator) udostępnia metodę `Classify()` służącą do przeprowadzenia klasyfikacji wektorów pewnego zbioru danych podawanego jako parametr tej funkcji. Takie obiekty (a właściwie ich metody) mogą zostać użyte wiele razy dla różnych parametrów wejściowych bez konieczności ponownego konfigurowania, tworzenia oraz uruchamiania modelu.

Choć cała architektura systemu Intemi może wydawać się dość skomplikowana na pierwszy rzut oka, to jednak przy bliższym poznaniu widać jej dobrze przemyślaną logikę. Najprościej mówiąc system Intemi to „piaskownica z repozytorium zabawek", przy czym piaskownicą jest tutaj jądro systemu, zaś zabawkami są dostępne w systemie modele.

4.1.2. Wpływ architektury Intemi na architekturę pluginu

Z powodu modularnej architektury Intemi proces zbierania wyników algorytmów uczenia maszynowego, będących podstawą do przeprowadzenia każdego statystycznego testu oceny istotności różnic tych algorytmów, jest zupełnie niezależny od przeprowadzenia samej procedury testowej. Mówiąc konkretnie do pluginu testów statystycznych zostają dostarczone dwie próby oraz informacja o ich parowalności, czyli sposobie uzyskiwania wyników tych prób. Plugin nie ma możliwości ingerencji w dostarczone wyniki, ani nie ma możliwości samodzielnego pobierania wyników porównywanych metod. Cały proces zbierania wyników musi zatem zostać przeprowadzony przed powołaniem instancji pluginu.

Takie rozwiązanie może wydawać się nieco nietypowym, gdyż w pracach opisujących zastosowanie testów statystycznych w porównywaniu metod uczenia maszynowego raczej spotyka się odmienne spojrzenie na ten

problem. W pracy Diettericha [2] czy Kunchevy [3] znajdziemy opisy testów bazujące na sposobie pobierania wyników systemów uczących się. Dobrym przykładem jest tu chociażby test Diettericha omówiony w podrozdziale 3.2.5, wymagający wykonania pięciu powtórzeń dwukrotnej kroswalidacji. Oczywiście każdy test, opisany w tych pracach możliwy jest do wykonania za pomocą omawianego pluginu. Co więcej nie musimy ograniczać się jedynie do konkretnej metody resamplingu danych przy pobieraniu wyników. Dajemy tym samym wolną rękę użytkownikowi systemu co do trenowania oraz testowania badanych algorytmów.

4.2. Podział pluginu na podmoduły

W całej architekturze pluginu można wyróżnić cztery najważniejsze części składowe:

- zbiór testów statystycznych, czyli pakiet klas realizujących poszczególne testy;
- klasę `StatisticalTestsPlugin` pośredniczącą pomiędzy jądrem systemu a zbiorem testów;
- zbiór rozkładów, czyli pakiet klas dostarczających funkcje, pozwalające na wyliczenie prawdopodobieństw dla wyznaczanych statystyk;
- zbiór funkcji pomocniczych.

W kolejnych podrozdziałach zostaną szczegółowo omówione wszystkie wymienione części składowe pluginu.

4.2.1. Zbiór testów statystycznych

Wszystkie testy statystyczne dostępne w pluginie zebrane są w jeden pakiet o nazwie `Tests`. Każdy pojedynczy test realizowany jest przez jedną klasę. I tak na przykład test McNemara realizowany jest za pomocą klasy `McNemarTest`, zaś sparowany test t-Studenta za pomocą klasy `PairedTTest`. W tabeli 4.1 przedstawione zostały wszystkie zaimplementowane w chwili obecnej testy istotności oraz nazwy klas, które je realizują.

Nazwa testu	Nazwa klasy
Testy sparowane	
McNemara	`McNemarTest`
różnicy dwóch proporcji	`DifferenceOfTwoProportionsTest`
sparowany t-Studenta	`PairedTTest`
kolejności par Wilcoxona	`WilcoxonTest`
Testy niesparowane	
χ^2	`ChiSquareTest`
F	`FTest`
niesparowany t-Studenta	`UnpairedTTest`
Manna-Whitneya-Wilcoxona	`MannWhitneyWilcoxonTest`

Tabela 4.1: Nazwy klas realizujących odpowiednie testy.

Podstawową rzeczą jaką należy nadmienić w celu ułatwienia zrozumienia architektury pluginu jest fakt, iż żaden pojedynczy test nie jest widziany jako model w systemie Intemi (tj. nie implementuje interfejsu `IModel`). Modelem widzianym przez jądro systemu jest jedynie klasa `StatisticalTestsPlugin`,

która zostanie omówiona w kolejnym podrozdziale. Zatem sam zbiór testów jest pewnym wewnętrznym elementem pluginu. Jądro nie wywołuje bezpośrednio konkretnego testu, a jedynie używa odpowiedniej metody dostarczanej przed klasę `StatisticalTestsPlugin`.

Każdy pojedynczy test musi implementować interfejs `ITest`. Interfejs ten zapewnia implementację metody `Run()`, służącej do uruchomienia testu oraz właściwość `TestOutput`, dzięki której uzyskiwany jest obiekt typu `ITestOutput` zawierający szczegółowe informacje na temat wyniku oraz przebiegu procedury testowej.

Wszystkie testy (a dokładniej mówiąc klasy je realizujące) opisane są przy użyciu atrybutu `TestInfoAttribute`. Za jego pomocą możemy uzyskać (statycznie) trzy podstawowe informacje o teście: nazwę testu, stosowalność dla prób sparowanych albo niesparowanych oraz możliwość badania hipotez kierunkowych. Za pośrednictwem tego atrybutu możemy, bez konieczności powoływania do życia obiektu klasy realizującej test, sprawdzić, czy dany test jest stosowalny do danych jakie aktualnie zamierzamy badać. Na takie dane składają się zarówno próbki – wektory wyników uzyskane w procesie testowania klasyfikatorów, jak również parametry przekazywane za pomocą klasy konfiguracyjnej klasy `StatisticalTestsPlugin`.

4.2.2. Klasa **`StatisticalTestsPlugin`**

Głównym elementem pluginu jest klasa `StatisticalTestsPlugin`, pośrednicząca pomiędzy jądrem systemu a zbiorem testów statystycznych. Jak to zostało wspomniane w poprzednim podrozdziale (4.2.1), klasa ta widziana jest w systemie Intemi jako model, to znaczy implementuje interfejs `IModel`. Jest ona odpowiedzialna za wybranie odpowiednich testów wobec rozważanego problemu oraz za wykonanie wskazanego przez użytkownika

testu. Selekcja testów dokonywana jest na podstawie ustawionych parametrów dla tej klasy. Parametry ustawiane są przy użyciu klasy konfiguracyjnej o nazwie `StatisticalTestsPluginConfig`. Istnieje możliwość manipulacji następującymi dwoma parametrami wpływającymi na wspomnianą selekcję:

- `Paired` – określa parowalność prób, czyli wpływa na wybór pomiędzy testem sparowanym a niesparowanym; możliwe wartości boolowskie `true` i `false`;
- `HypothesisDirection` – określa kierunek hipotezy alternatywnej, czyli decyduje o wyborze pomiędzy testami kierunkowymi a niekierunkowymi; możliwe wartości wyliczeniowe: `FirstBetter`, `NotEqual` oraz `SecondBetter`.

Klasa `StatisticalTestsPlugin` uzyskuje listę wszystkich dostępnych w pluginie testów statystycznych za pomocą wywołania statycznej metody `GetAllTests()` należącej do klasy `TestList`. Następnie dla każdego testu znajdującego się na otrzymanej liście sprawdza, czy dany test spełnia warunki wyznaczone przez dwa powyższe parametry dostarczone przy użyciu klasy konfiguracyjnej (a właściwie interfejsu `IModelBase`). Wykorzystuje do tego celu statyczne informacje o teście opisane atrybutem `TestInfoAttribute`.

Wejście tego modelu zdefiniowane jest interfejsem `IPairedSamples` (dla prób sparowanych) lub też `ITwoSamples` (dla prób niesparowanych), w zależności od tego jakim rodzajem prób dysponujemy, czyli, mówiąc dokładniej, w zależności od tego w jaki sposób pobieraliśmy wyniki porównywanych algorytmów. Oczywiście wejście to jest nieodzownie związane z parametrem `Paired` opisanym powyżej.

Wyjście modelu z kolei definiuje interfejs `IStatisticalTestsPlugin`, za pomocą którego udostępniane są dwie bardzo ważne metody:

- `SuitableTests` – będąca w zasadzie właściwością, która zwraca listę testów, jakie można przeprowadzić dla aktualnych danych;
- `DoTest()` – uruchamiającą podany jako argument do tej funkcji test.

Oczywiście za pomocą metody `DoTest()` można wykonać jedynie taki test, który znajduje się na liście uzyskanej w wyniku wywołania `SuitableTests`. W przypadku próby uruchomienia testu z poza listy zostanie wygenerowany wyjątek.

Taka architektura pluginu udostępnia jądru systemu Intemi narzędzie do wykonywania testów pasujących do danych dostarczonych w procesie konfiguracji pluginu. Obiekt klasy `StatisticalTestsPlugin` jest zatem pewnym zamkniętym (ze względu na zmianę parametrów) obiektem, posiadającym listę odpowiednich testów oraz udostępniającym metodę do ich wykonywania. Taka enkapsulacja zapewnia duże bezpieczeństwo pluginu, bowiem żaden inny model nie może zmienić jego parametrów w trakcie jego pracy.

W przypadku, gdy chcemy przeprowadzić test(y) z innym zestawem parametrów wejściowych pluginu należy utworzyć nową, odpowiednio skonfigurowaną instancję klasy `StatisticalTestsPlugin`. Nowa instancja będzie oczywiście zawierała listę testów pasujących dla nowych danych.

Nie trudno zauważyć, iż takie rozwiązanie niesie za sobą kolejną istotną zaletę. Chodzi mianowicie o fakt umożliwienia użytkownikowi utworzenia kilku instancji pluginu w celu porównania wyników algorytmów uzyskanych za pomocą różnych metod ich testowania.

4.2.3. Zbiór rozkładów

Kolejnym istotnym elementem pluginu jest zbiór klas z rozkładami statystyk. Wszystkie klasy zebrane są w jeden pakiet o nazwie `Distributions`. Ów pakiet dostarcza funkcje, za pomocą których mamy możliwość wyznaczania prawdopodobieństw dla wyliczanych statystyk. W aktualnej wersji pluginu mamy do dyspozycji następujące rozkłady: normalny, χ^2, F-Snedecora, t-Studenta, Wilcoxona oraz Manna-Whitneya-Wilcoxona.

Do wyliczania odpowiednich funkcji wykorzystane zostały darmowe kody źródłowe znalezione w Internecie. Wszystkie rozkłady opakowane są odpowiednio nazwaną klasą należącą do pakietu `Distributions`. Przykładowo funkcje statystyczne dla rozkładu normalnego znajdują się w klasie o nazwie `Normal`, zaś rozkład χ^2 realizowany jest przez klasę `ChiSquare`. Takie rozwiązanie zapewnia ujednolicenie dostępu do rozkładów bez konieczności zmiany nazw w wykorzystywanych kodach źródłowych.

4.2.4. Zbiór funkcji pomocniczych

W tym pakiecie znajdują się różne statyczne klasy zawierające metody, które realizują pewne pomocnicze zadania. Docelowo część funkcji z tego pakietu ma znaleźć się w jądrze systemu lub jego dodatkach zgodnie z jednym z podstawowych założeń Intemi, a mianowicie maksymalnego oddzielenia pozaalgorytmicznych elementów od właściwej części algorytmu. Znajdziemy tu przykładowo funkcje do wyliczania wariancji lub średniej z próby, testowania powtarzających się założeń testów (takich jak normalność czy binarność prób), jak również inne pomocnicze funkcje wykorzystywane przez różne elementy pluginu.

4.2.5. Podsumowanie architektury pluginu

W tym podrozdziale przedstawiony zostanie diagram klas, który z pewnością ułatwi wyobrażenie całej architektury pluginu jak i powiązań z jądrem systemu. Interfejsy `IModel`, `IConfiguration` oraz `IOutput` należą do jądra systemu Intemi, zaś interfejs `ICloneable` pochodzi z FCL (Framework Class Library) środowiska .NET. Pozostałe klasy oraz interfejsy są częścią omawianego pluginu testów statystycznych.

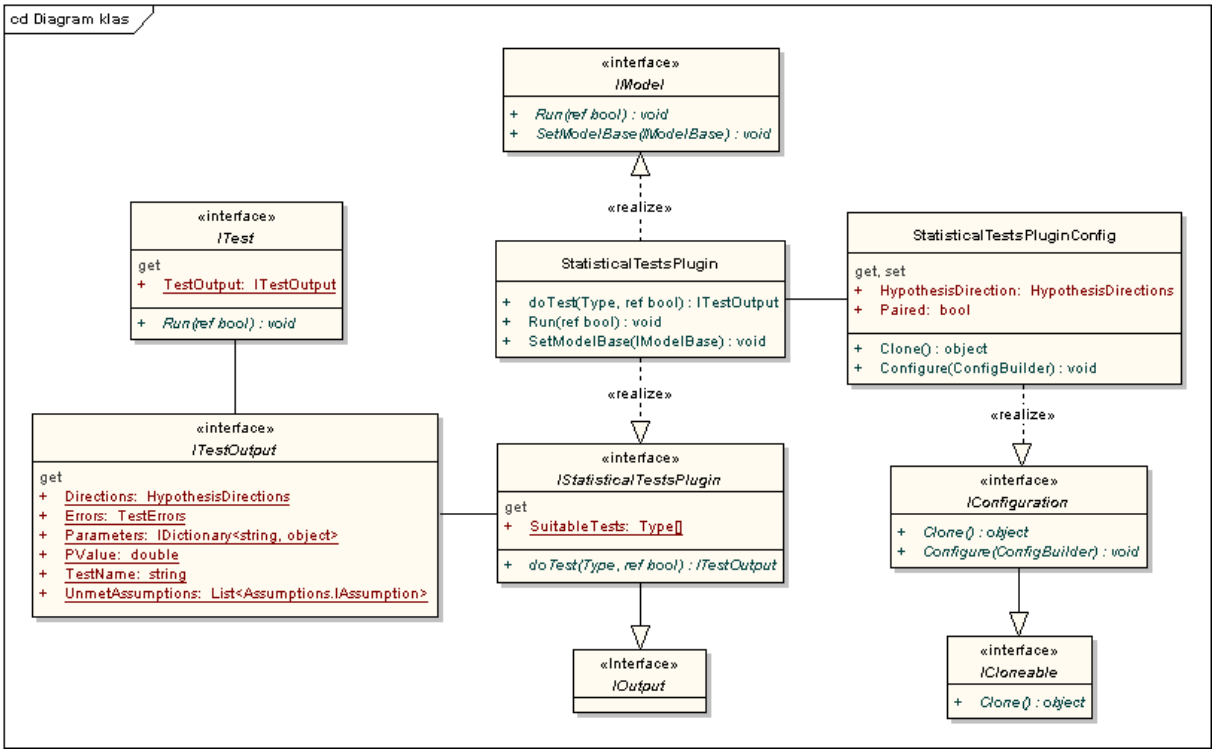


Diagram 4.1: Najistotniejsze klasy oraz interfejsy w pluginie.

4.3. Interakcje pomiędzy podmodułami

W poprzednim podrozdziale, zostały już zaprezentowane pewne aspekty interakcji pomiędzy podmodułami pluginu, które były konieczne do

zrozumienia jego ogólnej architektury. W tym podrozdziale przedstawiony zostanie ogólny schemat kolejno wykonywanych czynności, co powinno ułatwić zrozumienie zarówno architektury pluginu jak i wewnętrznego przepływu informacji pomiędzy jego podmodułami.

Ogólny przebieg działania pluginu przedstawia się w następujący sposób. W pierwszej kolejności następuje konfiguracja pluginu. Jak to zostało opisane w podrozdziale 4.2.2 konfiguracja ta polega na ustawieniu w klasie konfiguracyjnej `StatisticalTestsPluginConfig` dwóch parametrów: `Paired` i `HypothesisDirection` oraz powiązaniu odpowiednich wejść i wyjść klasy `StatisticalTestsPlugin` z odpowiednimi wejściami i wyjściami innych modeli. Oczywiście do wejścia pluginu musi zostać dowiązane wyjście pewnego modelu, które zdefiniowane zostało za pomocą interfejsu `IPairedSamples` lub `ITwoSamples`. Przykładem takiego modelu może być model kroswalidacji lub też model generatora próbek. Wyjście pluginu powinno być połączone z pewnym modelem komunikującym się z użytkownikiem systemu w celu wybrania odpowiednich testów do wykonania oraz wizualizującym wyniki wykonanych testów. Docelowo taka konfiguracja, a w szczególności łączenie odpowiednich modeli w systemie Intemi, ma być dokonywana w sposób graficzny.

Po odpowiednim skonfigurowaniu klasy `StatisticalTestsPlugin` jądro tworzy instancję tej klasy. W utworzonym obiekcie zostaje ustawiony model bazowy za pomocą metody `SetModelBase()` oraz wywołana zostaje metoda `Run()`, w której następuje przeprowadzenie selekcji testów. Po zakończeniu metody `Run()` mamy gotowe narzędzie do wykonywania testów statystycznych odpowiednich dla dostarczonych danych w procesie konfiguracji.

Model korzystający z wyjścia modelu `StatisticalTestsPlugin` uzyskuje listę testów możliwych do przeprowadzenia za pomocą właściwości

`SuitableTests`. Taką listę może przekazać użytkownikowi w celu dokonania przez niego wyboru testu, który zostanie następnie uruchomiony przy użyciu metody `DoTest()`. Metoda ta zwraca wynik w postaci obiektu typu `ITestOutput`, w którym znajdują się wszelkie informacje o przebiegu procedury testowej, jak również jej wynik. Dane te mogą zostać zaprezentowane użytkownikowi systemu. Oczywiście, o ile model komunikacji z użytkownikiem zapewni taką funkcjonalność, możliwe jest uruchomienie kolejnego testu z listy pasujących testów.

4.4. Wykorzystane narzędzia

Do wykonania pluginu, jak również całego systemu Intemi, została wykorzystana platforma .NET 2.0 wraz z Microsoft Visual Studio 2005. Został użyty bodajże jeden z najpopularniejszych języków wspieranych przez tą platformę, mianowicie język C#. Powodem wyboru takiego środowiska programistycznego przez głównych projektantów systemu Intemi dr N. Jankowskiego oraz dr K. Grąbczewskiego było ogromne wsparcie programistyczne zarówno od strony samego języka jak i narzędzia Visual Studio.

Ponadto, jak wspomniano w podrozdziale 4.2.3, w pakiecie rozkładów statystycznych do wyliczania odpowiednich funkcji zostały wykorzystane darmowe kody źródłowe znalezione w Internecie. Wszystkie kody napisane były w języku C/C++ i z tego powodu konieczna była ich konwersja do języka C#.

Największy problem stanowiło przełożenie biblioteki DCDFLIB z powodu jej dość obszernego rozmiaru – około 10 000 linii kodu. Biblioteka ta udostępnia wiele metod służących do wyliczania dystrybuant funkcji, ich

odwrotności oraz parametrów rozkładów dla następujących rozkładów statystycznych: beta, dwumianowy, χ^2, niecentralny χ^2, F, niecentralny F, gamma, ujemny dwumianowy, normalny, Poissona oraz t-Studenta.

Oprócz biblioteki DCDFLIB wykorzystane zostały jeszcze dwie zewnętrzne implementacje. Chodzi mianowicie o kod źródłowy programu do wyliczania progu istotności dla statystyki U z testu Manna-Whitneya-Wilcoxona, którego autorem jest Leon Avery oraz kod źródłowy funkcji obliczającej dokładny próg istotności dla statystyki W z testu Wilcoxona autorstwa Roba van Sona.

Rozdział 5

Porównanie wybranych testów na generowanych danych

W rozdziale tym zostanie zaprezentowana seria wykresów obrazująca bardzo ciekawy fenomen dużej niestabilności statystycznych testów istotności. Rozdział ten ma na celu uświadomienie faktu, jak często błędne są decyzje podejmowane przez testy o odrzuceniu hipotezy zerowej, mówiącej o nieistotności różnic. Wynikiem tego jest oczywiście złe oszacowanie poziomu istotności testów, który w rzeczywistości jest znacząco większy niż ten, wyznaczony przez prawdopodobieństwo uzyskania wartości statystyki testowej większej lub równej od wartości wyliczonej w procedurze weryfikacyjnej.

Wykresy przedstawiają zależność procentu odrzuconych hipotez zerowych w 1000 różnych losowań dwóch prób o rozkładzie normalnym, od odchylenia standardowego różnicy pomiędzy tymi rozkładami. W każdym z takich 1000 losowań parametry generowanych prób były identyczne.

Rozkłady sparowanych wyników zostały wygenerowane na podstawie następujących parametrów wejściowych:

- średnia pierwszego rozkładu: 0.7;
- standardowe odchylenie pierwszego rozkładu: 0.05;
- różnica pomiędzy średnimi rozkładów: 0.05;

- różnica pomiędzy standardowymi odchyleniami rozkładów: zmienna (na wykresach na osi odciętych).

Drugi rozkład jest zatem uzyskiwany z pierwszego poprzez dodanie poprawki o rozkładzie normalnym ze średnią 0.05, zaś standardowym odchyleniem zmieniającym się w taki sposób, aby przejść od wyraźnej statystycznej nieistotności (na wykresach po lewej stronie pionowej zielonej linii przerywanej), do wyraźnej statystycznej istotności (po prawej stronie linii).

Do wygenerowania niesparowanych wyników dokładności klasyfikacji zostały użyte następujące parametry:

- średnia pierwszego rozkładu: 0.7;
- standardowe odchylenie pierwszego rozkładu: 0.05;
- średnia drugiego rozkładu: 0.75;
- standardowe odchylenie drugiego rozkładu: zmienne (zależne od standardowego odchylenia różnicy rozkładów przedstawionego na wykresach na osi odciętych).

W tym przypadku na podstawie zmieniającego się odchylenia standardowego różnicy rozkładów wyznaczamy standardowe odchylenie drugiego rozkładu. Oczywiście zmiana standardowego odchylenia różnicy wykonywana jest w tym samym celu co dla prób sparowanych. Jako że rozkłady są niezależne, standardowe odchylenie drugiego rozkładu możemy wyliczyć według następującego wzoru:

$$\sigma_2 = \sqrt{\sigma_r^2 - \sigma_1^2} ,$$

przy czym σ_r oznacza odchylenie różnicy, zaś σ_1 i σ_2 są odchyleniami rozkładów odpowiednio pierwszego i drugiego.

Powyższe parametry są parametrami populacji generalnych, co oznacza, że parametry wygenerowanych rozkładów będą się nieco różnić od tych zadanych. Jest to sytuacja, z którą mamy do czynienia w realnych przypadkach – pobieramy próbę z jakiejś populacji i wcale parametry tej próby nie muszą być dokładnie takie jak parametry populacji źródłowej. Oczywiście wraz ze wzrostem rozmiaru próby powinny one znacznie mniej odbiegać od właściwych parametrów.

Dla uzyskania większej dokładności pomiarów testów, nie zostały obcięte ogony rozkładów dla wartości wykraczających poza przedział [0, 1]. Obcięcie ogonów wiązałoby się z zakłóceniem normalności rozkładów, które mogłoby w sposób znaczący (przy dużych odchyleniach standardowych) wpłynąć na końcowe rezultaty.

Hipoteza alternatywna, którą stawiamy, orzeka, że wartości drugiego rozkładu są istotnie większe od wartości pierwszego rozkładu. Na wykresach zieloną pionową linią przerywaną została oznaczona wartość standardowego odchylenia różnicy rozkładów, dla której uzyskiwana jest progowa wartość 0.05 rzeczywistego poziomu istotności. Taka wartość poziomu istotności jest jedną z najczęściej spotykanych wartości w literaturze uczenia maszynowego. Jeśli zadana istotność różnicy rozkładów (wyliczona z parametrów wejściowych) jest większa od poziomu 0.05, zaś istotność różnicy podawanej przez test jest mniejsza od tego poziomu, to w takim przypadku test podejmuje błędną decyzję (błąd typu I). To samo tyczy się odwrotnej sytuacji, czyli, gdy zadany poziom jest mniejszy od 0.05, a test twierdzi, że poziom ten jest większy (błąd typu II).

Na wykresach wraz ze spadkiem odchylenia standardowego różnicy rozkładów (a co za tym idzie wzrostem istotności różnicy) widoczny jest wzrost procentu odrzuceń hipotezy zerowej aż do maksymalnej wartości 100%. Wniosek z tego jest następujący: gdy mamy nieistotność różnic

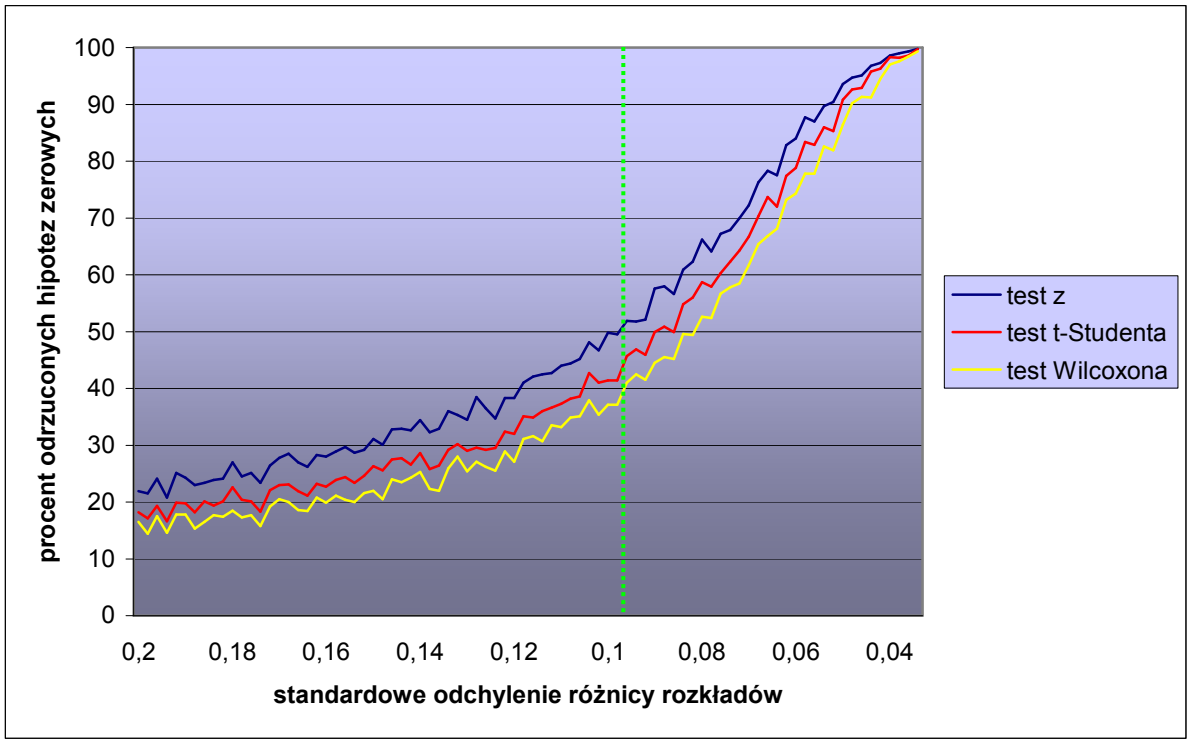
pomiędzy wynikami, to łatwo jest popełnić błąd odrzucając hipotezę o nieistotności, natomiast, gdy istotność ta jest wyraźna, to takie pomyłki w ogóle nie mają miejsca. Własność tą można w prosty sposób wytłumaczyć: jeśli mamy małe odchylenie standardowe różnicy rozkładów w stosunku do średniej tej różnicy, to rozkłady porównywanych grup w ogóle na siebie nie nachodzą. Z kolei przy bardzo dużym odchyleniu rozkłady nakładają się prawie całkowicie.

W przypadku, gdy procedura weryfikacyjna odrzuca hipotezę zerową na poziomie istotności $\alpha = 0.05$, wówczas na 100 przeprowadzonych testów co najwyżej 5 z nich powinno podjąć błędną decyzję o odrzuceniu prawdziwej hipotezy zerowej. Na podstawie poniższych wykresów nietrudno wysnuć wniosek, iż rzeczywisty poziom istotności jest znacznie większy niż ten, wyznaczany przez procedury testowe.

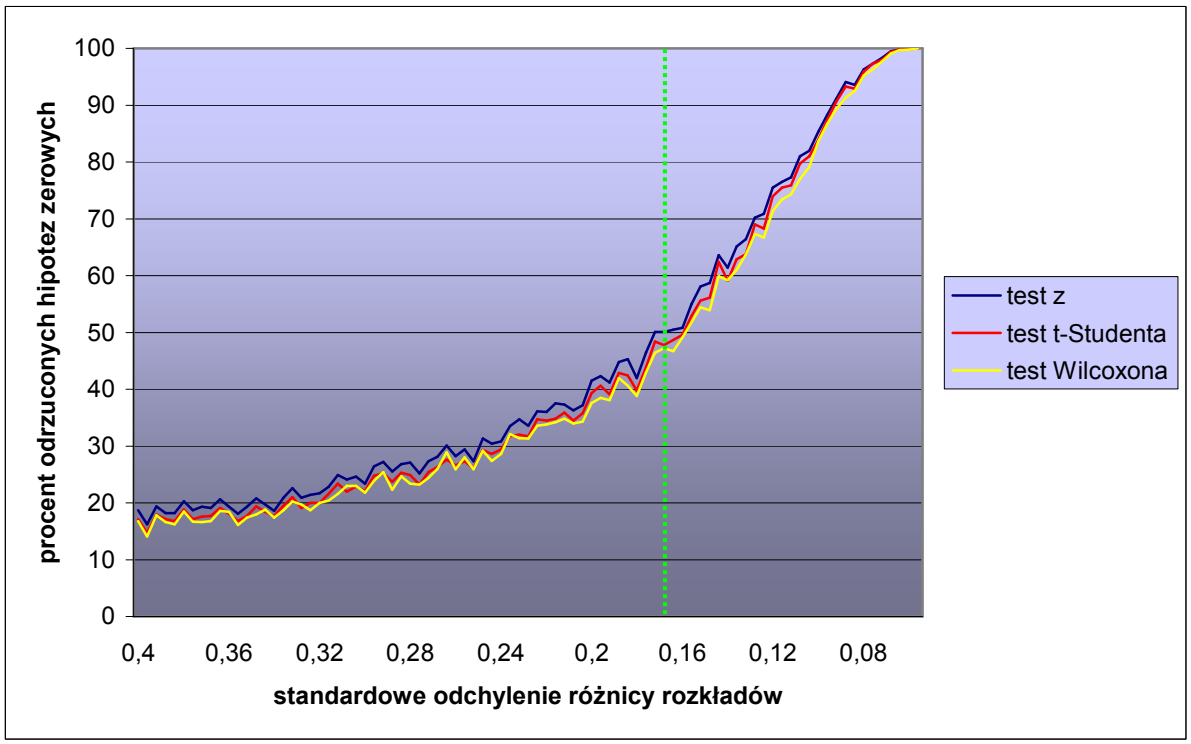
Ponadto należy zwrócić uwagę na fakt, iż w praktyce porównywania wyników systemów uczących się nie mamy raczej możliwości uzyskania wektorów z dokładnościami klasyfikacji o bardziej normalnych rozkładach niż te, które zostaną sztucznie wygenerowane. W wyniku tego poziom istotności różnicy będzie jeszcze gorzej przybliżany w realnych sytuacjach.

Bardzo ciekawym wnioskiem płynącym z tej analizy jest również fakt, iż nieparametryczne testy Wilcoxona oraz Manna-Whitneya-Wilcoxona zachowują się podobnie do testu t-Studenta w sensie liczby odrzucanych hipotez zerowych, a co za tym idzie uzyskiwanych błędów typu I oraz II. Jest to o tyle ciekawa obserwacja, że w literaturze statystycznej panuje ogólne przekonanie o wyższości tego parametrycznego testu nad jego nieparametrycznymi alternatywami, w sytuacjach, gdy badane próby mają rozkłady normalne.

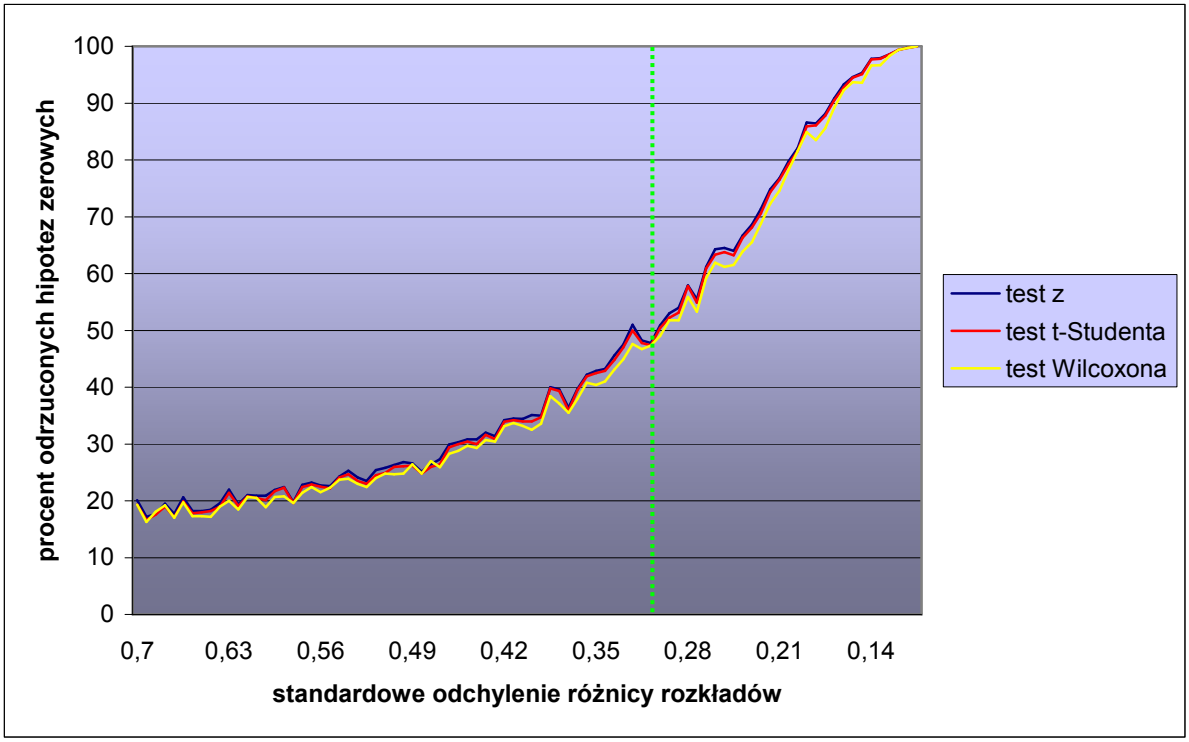
5.1. Wykresy dla testów sparowanych



Wykres 5.1: Procent odrzuconych hipotez dla prób wielkości $n = 10$.

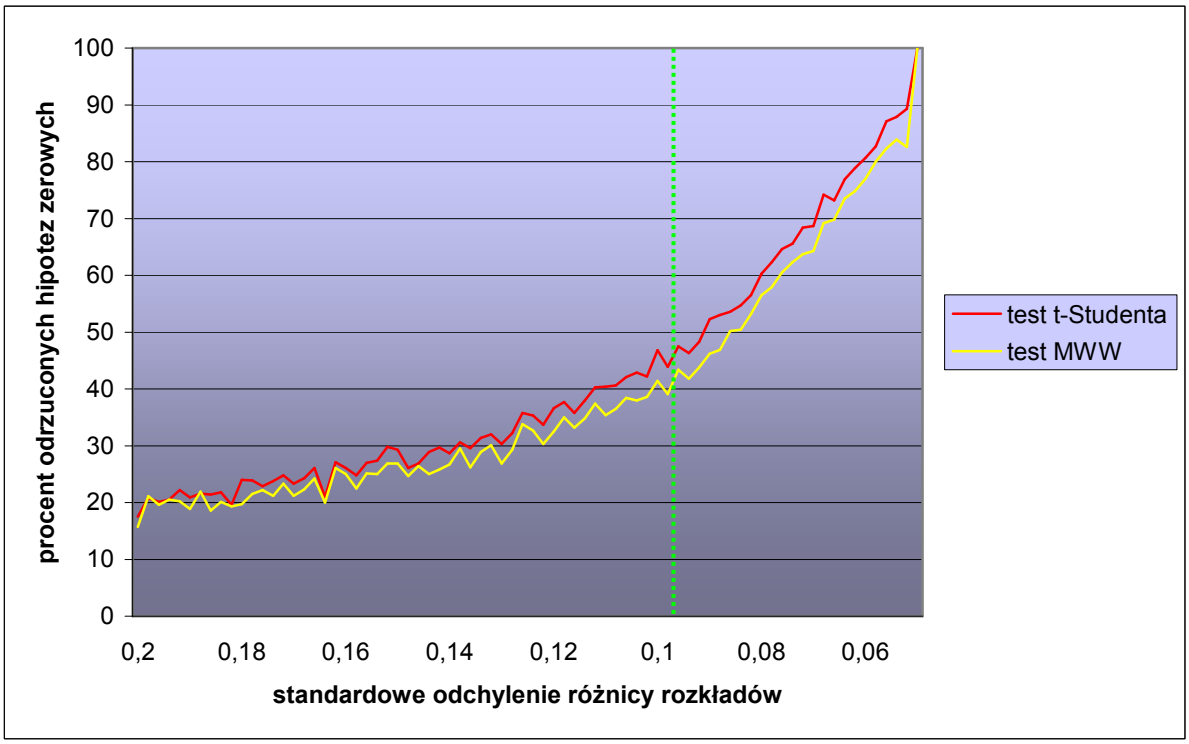


Wykres 5.2: Procent odrzuconych hipotez dla prób wielkości $n = 30$.

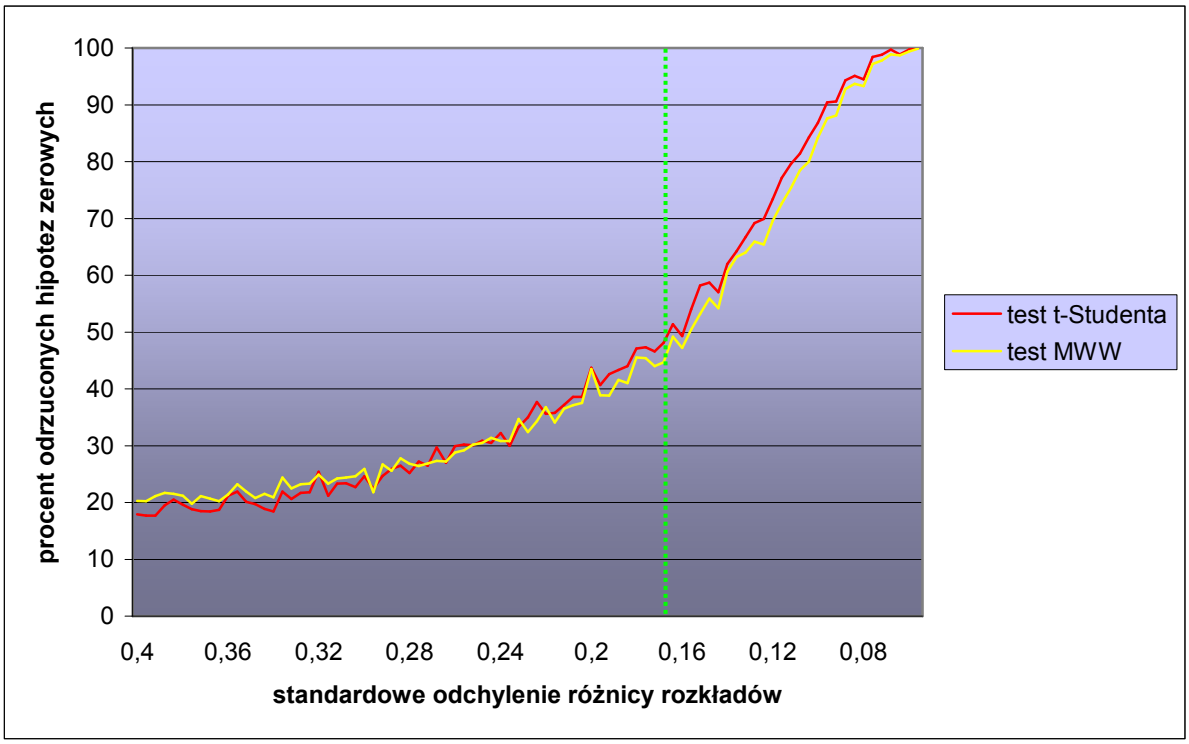


Wykres 5.3: Procent odrzuconych hipotez dla prób wielkości $n = 100$.

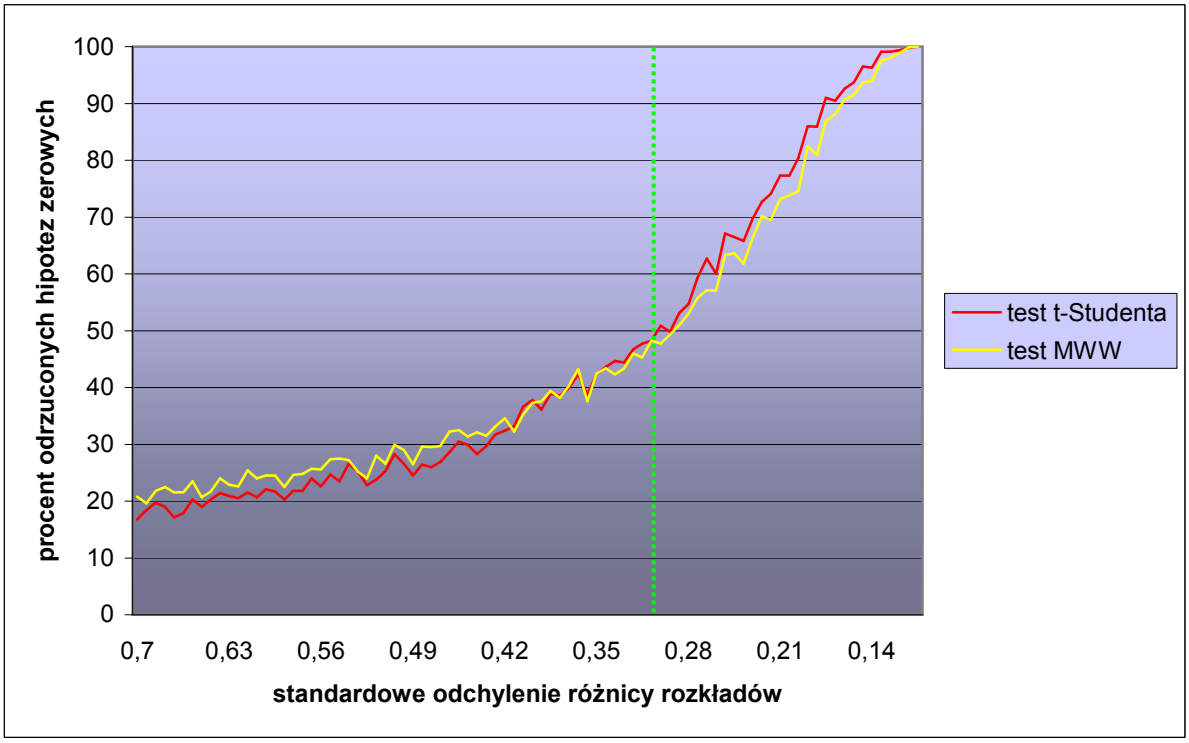
5.2. Wykresy dla testów niesparowanych



Wykres 5.4: Procent odrzuconych hipotez dla prób wielkości $n = 10$.



Wykres 5.5: Procent odrzuconych hipotez dla prób wielkości $n = 30$.



Wykres 5.6: Procent odrzuconych hipotez dla prób wielkości $n = 100$.

Rozdział 6

Zakończenie

W niniejszej pracy poruszony został problem porównywania wyników algorytmów uczenia maszynowego. Przyjrzeliśmy się statystycznemu narzędziu zwanemu testem istotności, które wydaje się być odpowiednim do zastosowania w tego typu zadaniach. Przedstawiony został ogólny schemat wykonywania testu istotności, wliczając w to krótkie omówienie sposobów pobierania wyników systemów uczących się oraz różnych problemów z tym związanych.

W dalszej części pracy zaprezentowano różne kryteria podziału zbioru testów statystycznych spotykane w literaturze, jak również szczegółową charakterystykę wybranych testów mających zastosowanie w ocenie istotności różnic w wynikach uzyskiwanych przez algorytmy uczenia maszynowego.

Omówiony został również prototypowy plugin do wykonywania testów statystycznych w powstającym obecnie nowym systemie do inteligentnej analizy danych o nazwie Intemi. Z powodu tego, iż system znajduje się ciągle w fazie rozwojowej, zaprezentowany plugin może nieco różnić się od jego finalnej wersji. Tym niemniej wszelkie kluczowe aspekty pluginu, jak również całego systemu Intemi, zostały zobrazowane w Rozdziale 4.

Architektura pluginu z założenia miała być na tyle uniwersalna, aby możliwe było łatwe rozbudowywanie go o nowe testy statystyczne. Udało się

tego dokonać poprzez użycie odpowiedniego łącznika pomiędzy zbiorem testów a jądrem systemu Intemi. Dodanie nowego testu statystycznego sprowadza się do utworzenia jednej klasy implementującej zaledwie dwie metody (dokładniej jedną metodę oraz jedną właściwość). Takie rozwiązanie jest w pełni zgodne z najważniejszymi założeniami nowego systemu, mianowicie modularnością oraz prostotą rozbudowy środowiska.

Ponadto w Rozdziale 5. przedstawione zostało krótkie porównanie wybranych testów statystycznych na sztucznie wygenerowanych danych o rozkładzie normalnym. Miało ono na celu zobrazowanie dużej niestabilności testów oraz niezbyt dokładnego oszacowania w nich poziomu istotności.

Wobec szybko rozwijającej się dziedziny Inteligencji Obliczeniowej oraz powstającej coraz to większej liczby metod uczenia maszynowego, niedługo trudno będzie się obejść bez jakiegoś sposobu porównywania wyników uzyskiwanych przez takie metody. Jako, że testy istotności są szeroko stosowanym narzędziem w różnych gałęziach nauki, nawet tak znaczących jak medycyna, można przypuszczać, iż niebawem żaden system do analizy danych wykorzystujący algorytmy uczenia maszynowego nie obędzie się bez takiego narzędzia. Wydaje się, że w chwili obecnej nie istnieje inny mechanizm lepiej radzący sobie z problemem oceny różnic w wynikach uzyskiwanych przez różne systemy uczące się.

Bibliografia

1. J. Greń. *Statystyka matematyczna. Modele i zadania*, Wydanie czwarte. Wydawnictwo naukowe PWN, Warszawa, 1975.

2. T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 7(10):1895–1924, 1998.

3. L. I. Kuncheva. *Combining Pattern Classifiers. Methods and Algorithms*. John Wiley & Sons, 2004.

4. R. Lowry. Concepts and Applications of Inferential Statistics. http://faculty.vassar.edu/lowry/webtext.html, 1999–2006.

5. R. Lowry. VassarStats: Web Site for Statistical Computation. http://faculty.vassar.edu/lowry/VassarStats.html, 1998–2006.

6. IFA Services: Statistics, Overview of tests. Institute of Phonetic Sciences (IFA), Faculty of Humanities, University of Amsterdam, The Netherlands. http://www.fon.hum.uva.nl/Service/Statistics.html.

7. T.-S. Lim, W.-Y. Loh, Y.-S. Shih. An empirical comparison of decision trees and other classification methods. Raport instytutowy, Department of Statistics, University of Wisconsin, Madison, Czerwiec 1997.

8. G. Dreyfus, I. Guyon. Assessment Methods, w: I. Guyon, S. Gunn, M. Nikravesh, L. Zadeh. Feature Extraction, Foundations and Applications, 65–88, Springer, 2006.

9. G. D. Garson. Quantitative Methods in Public Administration, Tests for Two Dependent Samples. PA 765, Quantitative Research in Public Administration (NCSU). http://www2.chass.ncsu.edu/garson/pa765/mcnemar.htm.

10. H. Motulsky. Intuitive Biostatistics. Oxford University Press Inc. http://www.graphpad.com/www/book/Choose.htm, 1995.

11. Module 28: Choosing a significance test. The International Development Research Centre, Canada. http://www.idrc.ca/en/ev-56462-201-1-DO_TOPIC.html.

12. Elektroniczny Podręcznik Statystyki PL. StatSoft, Kraków. http://www.statsoft.pl/textbook/stathome.html, 2006.

13. K. Grąbczewski. Zastosowanie kryterium separowalności do generowania reguł klasyfikacji na podstawie baz danych. Rozprawa doktorska, Warszawa, 2003.

14. E. W. Weisstein. *MathWorld* - A Wolfram Web Resource. http://mathworld.wolfram.com.

15. Wikipedia. http://wikipedia.org.

16. *NIST/SEMATECH e-Handbook of Statistical Methods*, http://www.itl.nist.gov/div898/handbook, 2003-2006.

17. D. Michie, D. J. Spiegelhalter, C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Elis Horwood, London, 1994.

18. J. Lani. Statistics Solutions, Inc. Clearwater, Florida, USA. http://www.statisticssolutions.com, 1996-2006.

19. K.-J. Lui. Notes on Testing Equality in Dichotomous Data with Matched Pairs. *Biometrical Journal*, 43: 313-321, 2001.

20. C. E. Lunneborg. *Data Analysis by Resampling: Concepts and Applications*. Duxbury Press, CB, 2000.

21. G. H. Givens, J. A. Hoeting. *Computational Statistics*. John Wiley & Sons, Hoboken, New Jersey, 2005.