Uniwersytet Mikołaja Kopernika
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Jakub P. Piątkowski

Analiza i rozwój metod selekcji cech dla dużych problemów klasyfikacyjnych

Praca magisterska
wykonana w Katedrze Informatyki Stosowanej
opiekun: **dr Norbert Jankowski**

Toruń 2006

Członkom Katedry Informatyki Stosowanej WFAiIS UMK,
promotorowi dr. Norbertowi Jankowskiemu
za okazaną pomoc

serdecznie dziękuję

Spis treści

Wstęp

Przedmiotem tej pracy jest zagadnienie selekcji cech na potrzeby klasyfikacji danych. Warto zatem na początek przedstawić pokrótce podstawowe pojęcia i konwencje stosowane w tej dziedzinie.

Zbiór danych

Dane mają postać zbioru *przypadków*, z których każdy opisany jest ciągiem d wartości ściśle określonych typów. Z tego powodu przypadki można traktować jako *punkty*, bądź *wektory*, w d-wymiarowej przestrzeni. Składowe tych wektorów noszą nazwę *cech* (stąd mowa o *przestrzeni cech*). Innymi często używanymi określeniami są *atrybuty* lub po prostu *zmienne*. Dodatkowo każdy wektor danych należy do określonej *klasy*. Pulę punktów o znanej przynależności klasowej nazywa się *zbiorem treningowym*.

Klasyfikacja

Zadanie *klasyfikacji* polega na określaniu przynależności wektorów danych spoza zbioru treningowego do jednej z dostępnych klas. Wymaga to rozróżnienia punktów należących do różnych klas, dlatego często wymiennie używa się terminu *dyskryminacja*. *Dokładność klasyfikacji* to ułamek liczby wszystkich klasyfikowanych przypadków, których klasy zostały określone poprawnie. Budowanie na podstawie zbioru treningowego wewnętrznych reprezentacji modeli używanych później do klasyfikacji nosi nazwę *procesu uczenia*.

Proces ten może polegać na oszacowaniu, na podstawie punktów treningowych, parametrów rozkładów danych dla różnych klas, albo na znalezieniu w przestrzeni cech kierunku, który zapewnia dobrą separację przypadków należących do różnych klas. Jego wynikiem może być również zestaw (zagłębionych) reguł w postaci: „jeśli wartość składowej f wektora $\mathbf{x}$ jest większa

niż γ, to należy on do klasy c". Ogólnie mówiąc, proces uczenia polega na określeniu na podstawie ograniczonej liczby wektorów tworzących zbiór treningowy pewnych uniwersalnych własności, wspólnych dla wszystkich przypadków z poszczególnych klas, co pozwala na *generalizację*, czyli skuteczną klasyfikację nowych danych.

W celu poprawnej oceny rozkładu klas w przestrzeni cech niezbędne jest dostatecznie gęste jej spróbkowanie. Nie da się ściśle określić, co to znaczy „dostatecznie gęste", ale można pokazać, że wraz ze wzrostem liczby wymiarów całkowita ilość wektorów potrzebna do zapewnienia tej samej gęstości próbkowania rośnie eksponencjalnie w funkcji d. Z tego powodu niebagatelną rolę w klasyfikacji odgrywają metody służące do zmniejszania wymiarowości danych.

Selekcja i ekstrakcja cech

Selekcja cech polega na wyborze, pod kątem przydatności do dalszego wykorzystania w klasyfikacji, niektórych atrybutów opisujących dane, a odrzuceniu innych. Jest ona zawsze rozpatrywana w kontekście późniejszych zadań; nie można ocenić jej skuteczności w oderwaniu od wyników metody klasyfikacyjnej, która korzysta z wyselekcjonowanych zmiennych. Najczęściej budowane są złożone modele, w skład których wchodzić może jeden lub więcej algorytmów selekcji oraz co najmniej jeden klasyfikator.

Ekstrakcja to budowanie nowych cech poprzez liniową lub nieliniową kombinację cech oryginalnych. W odróżnieniu od selekcji, gdzie celem jest zawsze uzyskanie pewnego podzbioru wszystkich atrybutów, wymiarowość przestrzeni będącej wynikiem ekstrakcji może być mniejsza, taka sama, a nawet większa niż wymiar przestrzeni startowej.

Ocena modeli

Wykorzystanie danych zgromadzonych w postaci zbioru treningowego polega na znalezieniu odpowiedniego klasyfikatora, przeprowadzeniu procesu uczenia z użyciem wszystkich dostępnych danych, a następnie na jego zastosowaniu do określania klas nowych przypadków.

Kryterium wyboru najlepszego modelu klasyfikacyjnego jest spodziewana dokładność klasyfikacji na danych niedostępnych podczas uczenia, dla których nie są znane etykiety klas. Oczywiście nie da się jej wyznaczyć, ponieważ nie sposób stwierdzić, które punkty zostały sklasyfikowane właściwie. Można

ją jednak oszacować testując model przy użyciu części zbioru treningowego nie używanego w procesie uczenia (*zbiór testowy*).

Jeszcze lepiej użyć w tym celu procedury *q-krotnej krosswalidacji* (ang. *q-fold crossvalidation* = qCV, np. 10CV). Przebiega ona w ten sposób, że zbiór treningowy D dzielony jest na q części $D_1, \ldots, D_q$. Następnie dla każdej części D_j klasyfikator jest uczony na przypadkach należących do reszty zbioru $(D \setminus D_j)$, a wektory należące do D_j służą jako zbiór testowy. W ten sposób otrzymuje się ciąg wartości, z którego można wyznaczyć nie tylko średnią dokładność, ale i jej wariancję.

Cele pracy

Celem tej pracy był przegląd różnych typów algorytmów selekcji cech oraz sposobów ich łączenia. Szczególny nacisk położony został na metody rankingowe, jako jedyne nadające się do dużych zbiorów danych. Zaproponowano również nową metodę tego typu, opartą o analizę czynników głównych (PCA). Skuteczność tego i innych algorytmów sprawdzona została przy wykorzystaniu zbiorów z konkursu NIPS Feature Selection Challenge, zawierających dużą liczbę atrybutów, a w niektórych wypadkach również niemałą ilość przypadków.

Rozdział 1

Selekcja cech

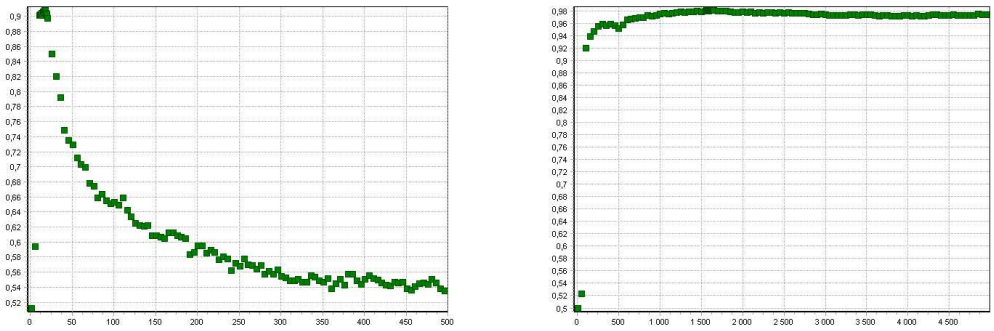
1.1 Cele selekcji cech

Podstawowym celem selekcji cech na potrzeby dyskryminacji jest zwiększenie dokładności algorytmu klasyfikującego. Jest to możliwe, jeżeli nie wszystkie cechy niosą istotną informację. Nie musi to znaczyć, że zmienne te są źle określone, pozbawione jakiejkolwiek wartości czy błędnie zmierzone; są one po prostu niezwiązane z rozpatrywanym w danym momencie zadaniem. W takim przypadku mówi się o braku korelacji z klasą, a cechy takie są często nazywane *zaszumionymi*. Rozkłady wartości takich atrybutów są niemal identyczne dla wektorów należących do różnych klas.

Warto podkreślić, że przydatność zmiennych zależy od wyboru klasyfikatora. Cechy zaszumione, szczególnie duża ich liczba, nie tylko nie pomagają, ale często mogą negatywnie wpływać na zdolności dyskryminacyjne niektórych modeli. W takich przypadkach maksimum dokładności osiągnąć można dla pewnego podzbioru cech[1]; dodawanie kolejnych zmiennych pogarsza wyniki (patrz rys. 1.1(a)).

Odmienną sytuację rozpoznać można na rysunku 1.1(b). Oto największa dokładność osiągnięta zostaje dla pewnej określonej liczby cech i utrzymuje się na podobnym poziomie również dla większych podzbiorów. Oznaczać to może jedną z dwóch sytuacji. Albo dodawane są cechy zaszumione, które jednak nie zakłócają pracy modelu, albo są to cechy równie istotne, jak znalezione wcześniej, nie wnoszące dodatkowej informacji. Takie *nadmiaro-*

[1]Zaznaczyć należy, że potencjalnie nie wszystkie metody selekcji znajdą podzbiór dający jakiekolwiek maksimum.



(a) (b)

Rysunek 1.1: Dokładność klasyfikacji (oszacowana przez krosswalidację) w funkcji liczby cech wybranych przez algorytm selekcji. Kształt zależności wskazuje na obecność cech zaszumionych (a) oraz nadmiarowych (b).

we zmienne mogą być silnie skorelowane z innymi, a nawet mogą to być po prostu ich przeskalowane odpowiedniki. Podobny efekt zaobserwować można dla cech o bardzo małej, czy wręcz zerowej wariancji.

Znalezienie pewnego podzbioru wszystkich cech umożliwiającego osiągnięcie większej, albo nawet takiej samej dokładności klasyfikacji, jak dla wszystkich zmiennych, jest sukcesem. Pozwala bowiem zidentyfikować wartości istotne dla zrozumienia rozważanego problemu. Pomaga również zbudować zredukowaną reprezentację danych, co z kolei prowadzi do skrócenia czasu obliczeń oraz oszczędności miejsca na dysku. Dalsze korzyści mogą płynąć z faktu, że do skutecznego rozpoznawania nowych przypadków wystarczy śledzić wartości jedynie niektórych parametrów problemu. Jako przykład podać można badania diagnostyczne konieczne do oceny stanu chorego.

1.2 Typy metod selekcji

1.2.1 Metody rankingowe

Najprostsze podejście do selekcji cech reprezentują *metody rankingowe*, nazywane też *filtrami*. Do stworzenia takiej metody potrzebny jest *indeks* (pewien współczynnik) określający dla każdej z osobna cechy jej jakość wedle przyjętego kryterium. Wyznaczanie wartości większości indeksów nie wymaga używania żadnego klasyfikatora, ale oczywiście pożądane jest, aby istniał

związek pomiędzy tymi wartościami, a dokładnością modeli, które pracują na danych po przeprowadzeniu selekcji cech.

Istnieje wiele sposobów budowania indeksów. Główne grupy algorytmów to [1]: metody oparte na korelacji wartości danej cechy z numerem (etykietą) klasy, odległościach pomiędzy ich rozkładami, miarach pochodzących z teorii informacji [2] i kryteriach używanych w drzewach decyzji. Wybrane metody rankingowe zostały szczegółowo omówione w rozdziale 2.

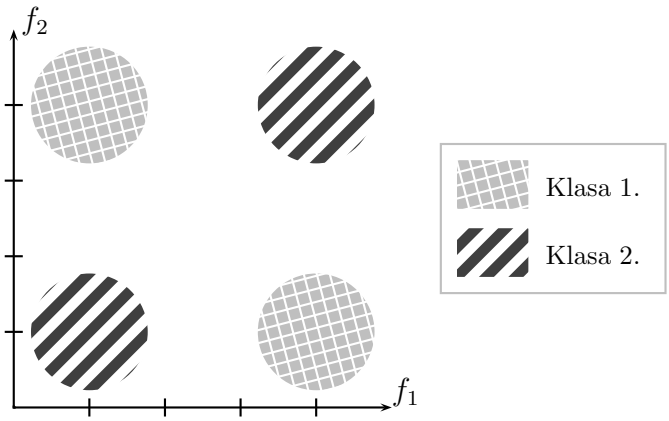
Wszystkie zmienne są oceniane przy pomocy indeksu, a następnie sortowane na podstawie jego wartości. Taką uporządkowaną listę cech nazywa się *rankingiem*. Selekcja polega na wyborze najlepszych (pierwszych), a odrzuceniu najgorszych (ostatnich) cech. Konieczny jest dodatkowy parametr, który wyznaczy miejsce odcięcia. Może to być ustalona liczba cech, które należy pozostawić, albo wartość indeksu oceniającego. Służy ona jako próg — nie są akceptowane atrybuty o niższych (lub wyższych) wartościach indeksu.

Największą zaletą selekcji rankingowej jest jej niska złożoność obliczeniowa. Jest to nierzadko jedyna klasa metod, jakie daje się w dostępnym czasie zastosować na dużych zbiorach danych. Z tego też powodu często wymaga się, żeby wyznaczanie wartości indeksu również miało niską złożoność.

1.2.2 Szukanie podzbiorów cech: wrappery

Metody rankingowe z definicji nie uwzględniają zależności pomiędzy cechami. Jest to podejście uzasadnione, jeśli rozpatrywać zmienne niezależne, ale w przypadku występowania korelacji pomiędzy cechami może okazać się niewystarczające. Rysunek 1.2 przedstawia rozkład danych, dla którego każdą z dwóch cech z osobna można by uznać za zaszumioną, ale uwzględnienie ich współdziałania daje szansę na poprawną klasyfikację. Widać zatem, że nie wszystkie cechy z dobrego podzbioru muszą być przydatne w oderwaniu od reszty. I odwrotnie, zmienne, które są najlepsze jeśli rozpatrywać je pojedynczo, niekoniecznie tworzą optymalny podzbiór. Mając do dyspozycji mechanizm oceny jakości całych zestawów cech można próbować znaleźć ciekawy z punktu widzenia klasyfikacji zbiór zmiennych uwzględniając zależności pomiędzy nimi.

W odróżnieniu od metod rankingowych, gdzie proces selekcji jest w zasadzie niezależny od wyboru korzystającego z jego wyników klasyfikatora, ocena podzbiorów cech jest najczęściej dokonywana przy użyciu pewnego modelu klasyfikacyjnego. Miarą jakości podzbioru jest dokładność klasyfika-



Rysunek 1.2: Przykładowy rozkład danych, dla którego żadna z dwóch cech nie daje możliwości odróżnienia wektorów klasy 1. od wektorów klasy 2., ale użycie obu pozwala poprawnie klasyfikować. Taki rozkład określa się jako XOR przez analogię do funkcji logicznej o tej samej nazwie.

tora, oszacowana przy użyciu krosswalidacji. Selekcja jest „owinięta" wokół tego modelu; stąd nazwa (ang. *wrapper* = opakowanie). Ten sam model używany jest w trakcie selekcji, jak i później, do klasyfikacji.

Oczywiście, znalezienie optymalnego[2] podzbioru jest w praktyce najczęściej niemożliwe ze względu na eksplozję kombinatoryczną. Dla d-wymiarowych danych istnieje $2^d - 1$ różnych, niepustych podzbiorów cech; zakładając z góry, że ma zostać wybranych k zmiennych, liczba ta zmniejsza się do $\binom{d}{k}$. To zbyt wiele żeby badać wszystkie dostępne możliwości i dlatego w wrapperach stosuje się algorytmy szukania.

Istnieje wiele strategii [3] zaprojektowanych w celu osiągnięcia rozsądnego kompromisu pomiędzy jakością znalezionego (nieoptymalnego) podzbio-

[2]Szukanie podzbioru optymalnego ze względu na wyniki krosswalidacji może prowadzić do fałszywej oceny jakości modelu [3].

Należy pamiętać o tym, że krosswalidacja jest jedynie oszacowaniem spodziewanej dokładności modelu na niewidzianych danych — w szczególności zależy od liczby podzbiorów (np. 10CV) i sposobu podziału zbioru treningowego. W przypadku kiedy proces uczenia jest sterowany jej wynikami, może to prowadzić do nadmiernego dopasowania (ang. *overfitting*) do tych parametrów (w klasyfikacji częściej jest mowa o nadmiernym dopasowaniu do danych — jest to inne zjawisko).

Aby uzyskać wiarygodne szacunki dotyczące klasyfikacji nowych przypadków, należy uczenie klasyfikatora umieścić w dodatkowej, zewnętrznej pętli krosswalidacji, albo co najmniej wydzielić zbiór testowy (patrz str. 7).

ru cech a zużytym czasem obliczeń. Rozwijają one z grubsza dwa podejścia: stopniowo dodają zmienne startując ze zbioru pustego (ang. *forward selection*), albo je odrzucają zaczynając ze wszystkimi dostępnymi cechami (ang. *backward selection*).

Uwzględnienie współdziałania cech odbywa się za cenę złożoności obliczeniowej, znacznie zwiększonej poprzez proces szukania. Stąd w wrapperach znajdują zastosowanie raczej szybkie klasyfikatory.

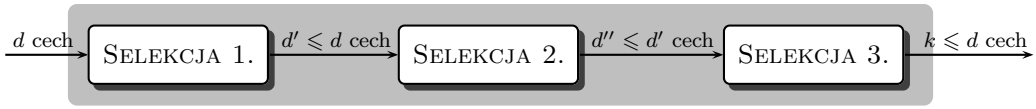
1.2.3 Metody wbudowane

Metody wbudowane (ang. *embedded methods*) korzystają z wewnętrznych reprezentacji wybranych klasyfkatorów, które w procesie uczenia dokonują pośrednio ocey przydatności cech, ich ważenia bądź wręcz selekcji. W [4] znaleźć można przykłady algorytmów opartych między innymi na dyskryminacji liniowej, sieciach neuronowych, SVM. W rozdziale 2.4 przedstawione zostały metody rankingowe oparte na drzewach decyzji, które podczas procesu uczenia również dokonują wyboru przydatnych cech.

1.3 Łączenie różnych selektorów

1.3.1 Różne modele

Zgodnie z zasadą, że nie ma jednego najlepszego selektora sprawdzającego się na każdym zbiorze danych, można próbować łączyć zalety różnych metod budując z nich złożone modele.

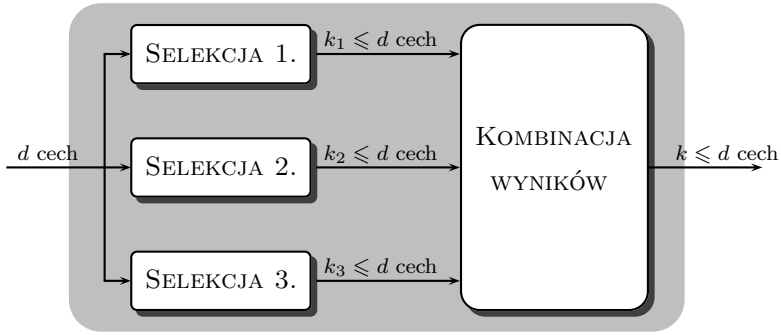


Rysunek 1.3: Schemat selekcji sekwencyjnej.

Najprostszym rozwiązaniem jest przetwarzanie sekwencyjne (rys. 1.3). Kilka algorytmów współdziała wtedy w ten sposób, że zbiór cech znaleziony przez jeden z nich jest traktowany jako punkt wyjścia dla następnego. Popularne jest tu użycie filtrów; każdy z nich może służyć do odfiltrowania

(stąd właśnie nazwa) cech zupełnie nieprzydatnych z punktu widzenia danego kryterium (indeksu). Inne zastosowanie to wstępna selekcja szybkim algorytmem (zwykle rankingowym) mająca na celu skrócenie, albo w ogóle umożliwienie, następujących po niej obliczeń metody o większej złożoności (wrappery, PCA).

Innym wyjściem jest stworzenie komitetu (rys. 1.4), czyli modelu, w którym wszystkie algorytmy składowe działają na całym zbiorze danych, a potem następuje kombinacja ich wyników. Na przykład, cechy wybrane przez wszystkie selektory dają niemal pewność, że okażą się przydatne. Jeżeli z kolei rozpatrywać cechy odrzucone przez wszystkie modele, można z dużym prawdopodobieństwem stwierdzić, że ich wartość dla klasyfikacji jest znikoma.



Rysunek 1.4: Schemat komitetu selekcji cech.

Uogólniając, dla komitetu n modeli można ustalić parametr m mówiący, że cecha jest wybierana, jeżeli wskazało na nią co najmniej m z n selektorów. Dodatkowo możliwe jest przyjęcie wag określających siłę głosu modeli składowych. Wygodnie jest, aby wagi $w_1, w_2, \ldots, w_n$ były liczbami całkowitymi, co jest równoważne dodaniu do komitetu $w_i - 1$ wirtualnych kopii selektora i. Zakres wartości m zwiększa się wtedy do $(0, \sum_{i=1}^{n} w_i)$. Komitety obu typów będą dalej skrótowo określane jako komitety „m z n" z lub bez wag.

Warto zauważyć, że opisane wyżej komitety można potraktować jako algorytmy rankingowe. Wartością indeksu oceniającego daną cechę jest tu suma wag selektorów, które ją wybrały, zaś m służy jako próg — cechy o niższych niż m wartościach indeksu są odrzucane.

Wadą komitetów „m z n" może być ich mała elastyczność — nie sposób zażądać od nich określonej liczby cech. Co więcej, uzyskany ranking jest

bardzo „gruboziarnisty", a to ze względu na małą dozwoloną liczbę różnych wartości indeksu. Na przykład, dla komitetu 3 selektorów indeks przyjmuje zaledwie 4 wartości: $\{0, 1, 2, 3\}$, a dla 4 selektorów z wagami (1,2,3,1) — 8 wartości: $\{0, 1, 2, 3, 4, 5, 6, 7\}$. Efekt jest taki, że wiele cech ma tę samą wartość indeksu, a co za tym idzie rezultatem działania powyższych metod może być jedynie (odpowiednio) 4 lub 8 różnych podzbiorów (włączając pełen zbiór cech dla $m = 0$) w zależności od doboru progu.

Receptą na powyższy problem w przypadku komitetów złożonych z filtrów może być wykorzystanie nie — jak zazwyczaj — zwracanych przez nie podzbiorów cech, lecz tworzonych przez te metody rankingów. Indeks dla takiego komitetu zbudować można jako sumę miejsc[3] danej cechy w rankingach poszczególnych algorytmów składowych. Przyjmuje on wtedy znacznie więcej wartości, co daje możliwość określenia z góry żądanej liczebności podzbioru. Ten rodzaj komitetów będzie dalej nazywany komitetami rankingowymi. Oczywiście tu także istnieje możliwość wprowadzenia wag.

Typ komitetu	Indeks	Zakres m	Sortowanie
„m z n" bez wag	$\sum_{i=1}^{n} s_i^{(d)}(f)$	$(0, n)$	malejąco
„m z n" z wagami	$\sum_{i=1}^{n} s_i^{(d)}(f) \cdot w_i$	$(0, \sum w_i)$	malejąco
rankingowy	$\sum_{i=1}^{n} s_i^{(p)}(f)$	(n, dn)	rosnąco
rankingowy z wagami	$\sum_{i=1}^{n} s_i^{(p)}(f) \cdot w_i$	$(\sum w_i, d \sum w_i)$	rosnąco

Tabela 1.1: Różne rodzaje komitetów selekcji cech jako szczególny przypadek metod rankingowych: odpowiadające im indeksy, zakresy ich wartości i sposób sortowania rankingów.

Spojrzenie na komitety jako na szczególny przypadek metod rankingowych pozwala na elegancką unifikację tych algorytmów. Niech dane będzie n modeli składowych: $s_1, \ldots, s_n$. Oznaczając przez $s_i^{(d)}(f)$ wartość binarną mówiącą, czy cecha f została wybrana (d=decyzja) przez i-ty selektor[4], a przez $s_i^{(p)}(f)$ — jej miejsce (p=pozycja) w rankingu tego modelu, można podsumować sposób tworzenia indeksu dla różnych rodzajów komitetów (patrz tab. 1.1). Oczywiście zakres parametru m (który jest warunkiem odcięcia nakładanym na wartość indeksu) zależy od typu komitetu. Dodatkowo od

[3]Sumowanie bezpośrednio wartości indeksu nie jest dobrą metodą ze względu na ich różny zakres.

[4]Wartość 1 oznacza, że tak, a 0 — że nie.

komitetów rankingowych można zażądać określonej liczby cech. Uważny czytelnik zauważy, że w komitetach „n z m" najlepsze cechy charakteryzują się wysoką wartością indeksu, a w komitetach rankingowych — niską. Stąd różnice w sortowaniu rankingu.

1.3.2 Wiele kopii tego samego modelu

Modele niestabilne są bardzo wrażliwe na zmiany w danych wejściowych, czyli niewielkie różnice w zbiorze treningowym powodują inny przebieg procesu uczenia i w efekcie inne decyzje podczas klasyfikacji. Za przykład służyć mogą drzewa decyzji. Takie algorytmy można stabilizować budując zespoły (ang. *ensemble*) złożone z takich samych modeli, ale uczonych w nieco innych warunkach.

Warto zaznaczyć, że opisane tu metody łączenia modeli w zespoły są ogólne i służyć mogą do poprawiania stabilności klasyfikatorów zarówno w trakcie selekcji cech opartej o takie metody (wrappery, metody wbudowane), jak i samych modeli dyskryminacyjnych, nie korzystających z wstępnej selekcji.

Istnieją dwa podejścia do budowania zespołów: równoległe i szeregowe. Przykładem pierwszego z nich jest *bagging* (= ang. *Bootstrap Aggregation*). Polega on na tym, że każda z kopii bazowego modelu jest uczona na innej próbce[5] zbioru treningowego losowanej z powtórzeniami (ang. *bootstrap sample*). Podejście szeregowe reprezentuje *Adaboost*. Każda kolejna kopia metody bazowej dodawana do zespołu ma za zadanie zająć się szczególnie dokładnie przypadkami, które są jak na razie źle klasyfikowane. Dokonuje się tego przez ważenie przypadków zbioru treningowego. Szczegółowe omówienie tych i innych metod budowania zespołów znaleźć można w [5].

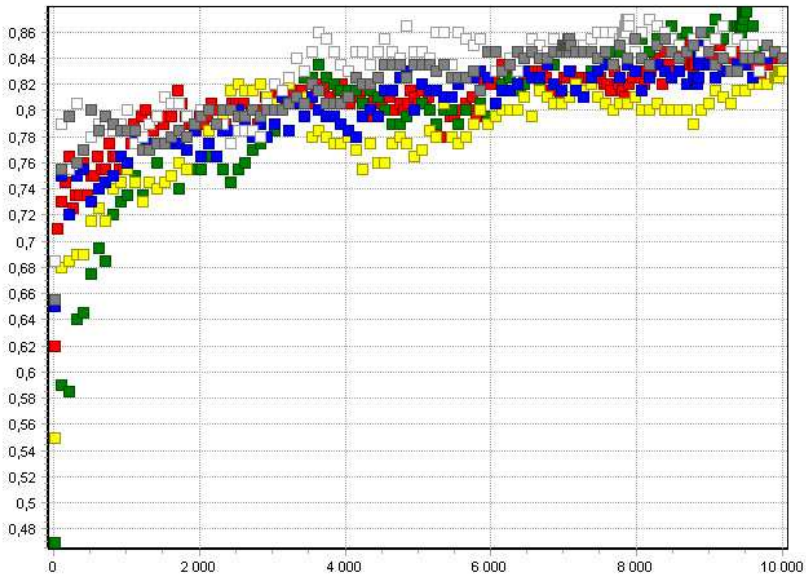
1.4 Podsumowanie, czyli którą metodę wybrać

Korzyści osiągnięte z selekcji cech zależą w głównej mierze od charakterystyki zbioru danych. Obecność silnie zaszumionych bądź skorelowanych zmiennych jest warunkiem powodzenia jakiegokolwiek algorytmu. Drugim w

[5]Liczebność takiej próbki jest zazwyczaj równa liczebności zbioru treningowego.

kolejności czynnikiem jest wybór metody klasyfikacji i to, jak reaguje ona na cechy o rozkładach prawie niezależnych od wyboru klasy.

Często w danym kontekście (zbiór danych i klasyfikator) większość selektorów zachowuje się w zbliżony sposób (patrz rys. 1.5). Za najlepszy w uznaje się wtedy ten model, który w połączeniu z rozpatrywanym algorytmem dyskryminacji osiąga dokładność lepszą od innych (chodzi o maksimum krzywej z rys. 1.5), albo zbliżoną, ale dla mniejszej liczby cech. Ostatnim kryterium może być wariancja wyników krosswalidacji dla maksymalnej dokładności — im mniejsza, tym lepiej.



Rysunek 1.5: Wiele selektorów (każdemu odpowiada inny kolor) w połączeniu z tym samym klasyfikatorem, na tym samym zbiorze danych.

Należy przypomnieć, że wyniki dokładności, o których mowa, to jedynie pewne oszacowanie mówiące, czego należy się spodziewać dla nowych przypadków, które klasyfikuje się przy użyciu wybranego ostatecznie modelu: selektor i klasyfikator po procesie uczenia na całym dostępnym zbiorze treningowym.

Warto podkreślić, że zarówno to, która metoda okaże się najefektywniejsza, jak i ewentualnie to, która będzie wyraźnie odstawać od reszty, zmienia się wraz z rozpatrywanym zbiorem danych. Ten sam algorytm może być

pierwszy na jednym zbiorze i ostatni na innym. Oznacza to, że warto używać wielu modeli, bo żaden nie daje gwarancji uniwersalnej użyteczności.

Rozdział 2

Wybrane metody rankingowe

Przed przystąpieniem do omawiania bardziej szczegółowo wybranych algorytmów selekcji cech warto wprowadzić konwencje zapisu używane w dalszej części pracy. Wektory danych oznaczane będą pogrubioną czcionką, a nawiasy kwadratowe stosowane będą do oznaczenia wyboru składowych, np. $\mathbf{x}[f]$ to wartość atrybutu f wektora $\mathbf{x}$. Zbiór wszystkich klas to C, a klasa numer i — $c_i \in C$. Zbiór treningowy (ozn. D) zawierać będzie n przypadków, z których n_i należeć będzie do klasy c_i (takie punkty tworzą zbiór D_{c_i}).

2.1 Współczynnik korelacji

Metoda współczynnika korelacji (ang. *Correlation Coefficient* = CC) bazuje na korelacji liniowej Pearsona. Dla dwóch zmiennych losowych a i b jest ona dana wzorem:

$$\rho_{ab} = \frac{E(ab) - E(a)E(b)}{S_a S_b}, \tag{2.1}$$

gdzie $E(\cdot)$ oznacza wartość oczekiwaną, a S_a i S_b są odchyleniami standardowymi. Mając dane zbiory wartości $\{a_1, \ldots, a_n\}$ i $\{b_1, \ldots, b_n\}$ takie, że pary (a_i, b_i) są ze sobą w pewien sposób powiązane (na przykład są to wielkości fizyczne mierzone podczas jednego pomiaru), można wyznaczyć korelację liniową Pearsona korzystając z:

$$\rho_{ab} = \frac{1}{S_a S_b} \frac{1}{n-1} \sum_{i=1}^{n} (a_i - \bar{a})(b_i - \bar{b}), \tag{2.2}$$

gdzie $\bar{a}$ i $\bar{b}$ są wartościami średnimi. Warto może przypomnieć, że S_a i S_b są dane przez (na przykładzie S_a):

$$S_a = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(a_i - \bar{a})^2},$$

co pozwala napisać:

$$\rho_{ab} = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{n}(b_i - \bar{b})^2}} \tag{2.3}$$

Współczynnik ρ_{ab} przyjmuje wartości z przedziału $[-1, 1]$, przy czym $|\rho_{ab}|$ określa siłę zależności, a znak — jedynie jej charakter (wprost/odwrotnie proporcjonalne). Współczynnik Pearsona określa, jak bardzo rozkład punktów (a_i, b_i) przypomina prostą. Dla $|\rho_{ab}| = 1$ istnieje ścisła zależność liniowa, natomiast dla niezależnych a i b zachodzi $\rho_{ab} = 0$.

Dla celów selekcji cech wyznacza się korelację wybranej cechy z klasą — jest to wartość indeksu, według którego tworzony jest ranking. Rolę zbioru a odgrywają wartości rozpatrywanego atrybutu dla wektorów zbioru treningowego, a zbiór b tworzą numery klas, do których należą poszczególne wektory. Indeks oparty na współczynniku korelacji liniowej Pearsona oznaczony będzie jako CC, czyli:

$$\mathrm{CC}(f) = \rho_{f,\mathrm{klasa}} \tag{2.4}$$

Ponieważ klasy mają znaczenie ściśle symboliczne i nie można stwierdzić, jaka jest ich kolejność, czy odległości pomiędzy nimi, nadawanie im numerów jest odbywa się w sposób dowolny. Niestety, nie pozostaje to bez wpływu na wartość współczynnika korelacji. Można to wykazać na prostym przykładzie. Niech dana będzie cecha f taka, że przydział dowolnego przypadku $\mathbf{x}$ do klasy określa formuła (β i γ są stałymi):

$$\mathbf{x} \in \begin{cases} \text{klasy nr 1. dla} & f \leqslant \beta \\ \text{klasy nr 2. dla} & f \in (\beta, \gamma) \\ \text{klasy nr 3. dla} & f \geqslant \gamma \end{cases}$$

Taki atrybut idealnie nadaje się do klasyfikacji; wartość $\rho_{f,\mathrm{klasa}}$ będzie w tym przypadku również stosunkowo wysoka. Ale jeśli zamienić ze sobą numery klas: 1. i 2., to wartość tego współczynnika zmaleje, mimo że przydatność cechy f pozostanie bez zmian.

Warto również podkreślić, że duże wartości współczynnika korelacji są równoznaczne z istnieniem zależności liniowej. Nie jest on przydatny do wykrywania zależności nieliniowych.

2.2 F-score

Współczynnik F-score (ozn. Fs) dla danych dwuklasowych mierzy separację wartości średnich dla klas. Ten indeks jest inspirowany kryterium Fishera, używanym w dyskryminacji liniowej. W metodzie Fishera szuka się kierunku (danego przez pewien wektor $\mathbf{w}$), na który następnie rzutowane są punkty danych, a który maksymalizuje:

$$K_F = \frac{\left|\bar{y}^{(1)} - \bar{y}^{(2)}\right|^2}{\left(S_y^{(1)}\right)^2 + \left(S_y^{(2)}\right)^2}, \tag{2.5}$$

gdzie $\bar{y}^{(1)}, \bar{y}^{(2)}$ są wartościami średnimi dla punktów z klas 1. i 2. (po rzutowaniu wektorom danych $\mathbf{x}$ odpowiadają pojedyncze wartości $y = \mathbf{w}^T\mathbf{x}$), a $S_y^{(1)}, S_y^{(2)}$ — odpowiednimi odchyleniami standardowymi. Znalezienie punktu podziału y_p pozwala klasyfikować nowe przypadki poprzez rzutowanie na kierunek $\mathbf{w}$ i sprawdzenie, po której stronie y_p znalazła się uzyskana wartość y (czyli $y > y_p$ albo $y < y_p$). Jest to równoznaczne z rozdzieleniem w przestrzeni oryginalnych cech obszarów odpowiadających różnym klasom — płaszczyzną.

Podobnie definiowana jest separacja wartości średnich dla pojedynczej cechy f, gdzie $\bar{f}^{(1)}$ i $\bar{f}^{(2)}$ są średnimi wartościami danego atrybutu w ramach klas 1. i 2., a $S_f^{(1)}$ i $S_f^{(2)}$ — odchyleniami standardowymi dla tych klas, czyli

$$\left.\begin{aligned} \bar{f}^{(i)} &= \tfrac{1}{n_i} \sum_{\mathbf{x}\in c_i} \mathbf{x}[f] \\ S_f^{(i)} &= \sqrt{\tfrac{1}{n_i - 1} \sum_{\mathbf{x}\in c_i} (\bar{f}^{(i)} - \mathbf{x}[f])^2} \end{aligned}\right\} \text{dla } i = 1, 2 \tag{2.6}$$

Wygląda ona wtedy następująco:

$$\mathrm{Fs}(f) = \frac{\bar{f}^{(1)} - \bar{f}^{(2)}}{S_f^{(1)} + S_f^{(2)}} \tag{2.7}$$

Podane kryterium można bezpośrednio stosować do problemów dwuklasowych, natomiast dla większej liczby klas konieczne jest zastosowanie jednego z dwóch podejść: rozpatrywanie klas parami (ang. *one vs. one*) lub każdej klasy przeciw wszystkim innym (ang. *one vs. rest*). W ten sposób otrzymuje się zbiór współczynników (dla każdej klasy lub pary klas), które należy zsumować celem uzyskania ostatecznej wartości indeksu F-score.

2.3 Relief

Algorytmy typu Relief oparte są na podobnym pomyśle, jak klasyfikator kNN, który szuka najbliższych sąsiadów klasyfikowanego punktu, a następnie zalicza go do najliczniej wśród nich reprezentowanej klasy.

Jeśli spojrzeć w ten sposób na selekcję, to należy promować te cechy, dla których sąsiedzi należący do tej samej klasy, co rozważany wektor treningowy, są blisko, a sąsiedzi z innych klas — daleko.

Podstawowa metoda wyznaczania indeksu Relief (ozn. Rf) działa w oparciu o próbkę μ przypadków zbioru treningowego (można wybrać $\mu = n$, czyli cały zbiór). Dla każdego punktu $\mathbf{x}$ z tego zbioru znajdowany jest najbliższy sąsiad z tej samej (ang. $hit = \mathbf{h}$) oraz z innej klasy (ang. $miss = \mathbf{m}$). Następnie dla każdej cechy f indeks jest powiększany o odległość $\mathbf{x}$ od $\mathbf{m}$ (liczoną w kierunku f) i pomniejszany o odległość od $\mathbf{h}$, co daje:

$$\mathrm{Rf}(f;\mu) = \frac{1}{\mu}\sum_{i=1}^{\mu} |\mathbf{x}_i, \mathbf{m}_i|_f - |\mathbf{x}_i, \mathbf{h}_i|_f, \tag{2.8}$$

gdzie $|\cdot,\cdot|_f$ jest odległością wzdłuż kierunku f, czyli $|\mathbf{x}_i, \mathbf{m}_i|_f \equiv |\mathbf{x}_i[f] - \mathbf{m}_i[f]|$.

Modyfikacja powyższej metody polegająca na uwzględnieniu κ sąsiadów z każdej z dwóch kategorii (z tej samej i innej klasy) zamiast jednego nosi nazwę ReliefF (ozn. RfF). Wtedy:

$$\mathrm{RfF}(f;\mu,\kappa) = \frac{1}{\mu\kappa}\sum_{i=1}^{\mu}\sum_{j=1}^{\kappa} |\mathbf{x}_i, \mathbf{m}_{ij}|_f - |\mathbf{x}_i, \mathbf{h}_{ij}|_f \tag{2.9}$$

Wzory (2.8) i (2.9) odnoszą się do cech o wartościach numerycznych. Dla atrybutów symbolicznych odległości $|\cdot,\cdot|_f$ należy zastąpić wartościami binarnymi: 1 — jeśli punkty mają taką samą wartość cechy f lub 0 — jeśli różną.

2.4 Drzewa SSV

Drzewa decyzji to przykład metod klasyfikacyjnych, które dokonują selekcji cech podczas procesu uczenia. Polega on na rekurencyjnym podziale przestrzeni cech, a co za tym idzie zbioru treningowego (najczęściej na dwie części — drzewa binarne) ze względu na wartość pewnego, wybranego w danym kroku, atrybutu. Celem jest uzyskanie w liściach drzewa podzbiorów, dla których prawie wszystkie wektory należą do tej samej klasy.

Zarówno wybór cech, jak i wartości używanych do podziału polega na optymalizacji pewnego kryterium, takiego jak zysk informacyjny (ang. *Information Gain*), indeks Gini, poziom błędu czy separowalność ze względu na punkt podziału (ang. *Separability of Split Value* = SSV). Ostatni z wymienionych współczynników opisany został poniżej.

Dla zbioru danych D punkt podziału s, który jest wartością cechy f definiuje dwa podzbiory będące wynikiem tego podziału: L i R, gdzie:

$$L(s, f, D) = \{\mathbf{x} \in D : \mathbf{x}[f] < s\}$$
$$R(s, f, D) = D \setminus L$$

Oznacza to, że podzbiór L zawiera wektory z D, dla których wartość cechy f jest mniejsza niż s, a zbiór R — wszystkie pozostałe. Dla atrybutów symbolicznych nie można zdefiniować punktu podziału; s jest wtedy pewnym podzbiorem dopuszczalnych wartości f, co daje:

$$L(s, f, D) = \{\mathbf{x} \in D : \mathbf{x}[f] \in s\}$$

Definicja zbioru R pozostaje dla takich cech bez zmian.

Niech, dla ustalonych f i s, $l(D)$ oznacza moc (liczbę elementów) zbioru $L(s, f, D)$, a $r(D)$ — moc $R(s, f, D)$. Wtedy kryterium SSV definiuje się jako:

$$\mathrm{SSV}(s, f, D) = 2 \sum_{c \in C} [l(D_c) \cdot r(D \setminus D_c)] - \sum_{c \in C} \min\left(l(D_c), r(D_c)\right). \quad (2.10)$$

Do budowy drzewa wykorzystywane są f i s maksymalizujące $\mathrm{SSV}(s, f, D)$.

Kryterium (2.10) można bezpośrednio użyć do oceny przydatności cech [6], na potrzeby selekcji. Taki model określa się jako jednopoziomowy (ang. *one-level*), ponieważ odpowiada on decyzji podjętej na jednym (pierwszym) poziomie drzewa. Jako wartość indeksu dla cechy f w metodzie rankingowej SSV przyjmowana jest wtedy największa wartość $\mathrm{SSV}(s, f, D)$ biorąc (teoretycznie) pod uwagę wszystkie punkty podziału dla danej cechy. Oczywiście w praktyce badane są jedynie niektóre możliwości, podobnie jak podczas budowy drzew.

Bardziej złożone algorytmy [6] wyznaczają wartość indeksu w oparciu nie o samo kryterium SSV, ale badając całe drzewa zbudowane przy jego użyciu. Można na przykład dla każdej cechy f zbudować oddzielne drzewo (w poszczególnych węzłach zmieniają się jedynie s) i na jego podstawie określić

przydatność tego atrybutu. Jest to metoda typu „drzewo na cechę" (ang. *tree for each feature*).

Innym podejściem [6] jest zbudowanie dla pełnowymiarowych danych jednego drzewa decyzji (stąd nazwa: „pojedyncze drzewo", ang. *single tree*), a następnie przeprowadzenie jego przycinania (ang. *pruning*), czyli stopniowego odrzucania najmniej wartościowych węzłów (liści). Atrybuty są oceniane według kolejności ich usuwania z drzewa. To znaczy, że cechy niewykorzystane do jego budowy otrzymują wartość indeksu równą 0, atrybut odrzucony jako pierwszy — 1, a każdy następny o 1 więcej. Wadą tego podejścia jest to, że dla wysokowymiarowych danych drzewa decyzji często wykorzystują jedynie niewielką część wszystkich cech, co ogranicza możliwości tej metody. Pozostałe atrybuty nie zostają bowiem ocenione i nie ma podstaw do ich posotrowania. Efekt jest taki, że selekcja rankingowa tego typu wybiera maksymalnie taką liczbę cech, jaka została użyta do budowy drzewa.

Rozdział 3

Selekcja w oparciu o PCA

3.1 Principal Component Analysis

Analiza czynników głównych (ang. *Principal Components Analysis* = PCA) to jedna z najpopularniejszych metod stosowanych w celu wizualnej oceny występujących w danych zależności, klastrów itp. (często ang. *Exploratory Data Analysis* = EDA). Używa się jej do zmniejszenia wymiarowości danych przy zachowaniu możliwie największej części wariancji w nich zawartej. PCA bywa też stosowana jako metoda ekstrakcji[1] cech dla algorytmów klasyfikujących. Warto tutaj zaznaczyć, że w przeciwieństwie do większości metod selekcji cech jest to metoda uczona bez nadzoru, co oznacza, że nie jest w niej brana pod uwagę przynależność wektorów danych do poszczególnych klas.

Celem PCA jest znalezienie w przestrzeni danych kierunków o pewnych określonych własnościach. Noszą one nazwę *składowych głównych* lub *kierunków głównych* (ang. *principal components*) i tworzą nową bazę, w której można zapisać wektory danych. Należy wyraźnie podkreślić, że nie ma w tym procesie miejsca żadne skalowanie — zostają zachowane wszystkie odległości i kąty oraz wymiar przestrzeni. Żąda się, żeby nowa baza była ortonormalna oraz aby spełnione były następujące warunki:

- Dla danych zapisanych w bazie kierunków głównych cechy są nieskorelowane (czyli mają zerową kowariancję — patrz równanie (3.3)).
- Składowe główne są uporządkowane według malejącej wariancji.

[1] Różnica między selekcją a ekstrakcją cech — patrz str.6.

Ograniczenie wymiarowości danych osiąga się poprzez odrzucenie kierunków głównych, dla których dane mają małą wariancję i pozostawienie jedynie pewnej liczby pierwszych (czyli „najlepszych") głównych składowych. Istnieją pewne heurystyczne reguły określania tej liczby, ale w praktyce kryterium wyboru jest tu zupełnie dowolne.

Wykorzystanie pierwszej z wymienionych własności PCA do znajdowania kierunków głównych zostało szczegółowo opisane w rozdziale 3.2. W podręczniku Webba [7] znaleźć można inne podejście[2], bazujące na drugiej własności PCA. Pierwsza składowa główna jest znajdowana jako kierunek, w którym dane mają największą wariancję. Natomiast każda kolejna jest dobierana w taki sposób, żeby odpowiadała za jak największą część pozostałej wariancji i była prostopadła do kierunków głównych znalezionych wcześniej.

Uważny czytelnik zauważy w tym miejscu, że PCA jest wrażliwa na skalowanie danych wejściowych. W szczególności, jeżeli zakres zmienności jednej ze zmiennych jest znacząco większy niż pozostałych, to zdominuje ona pierwszy kierunek główny, ponieważ odpowiada za znaczącą część wariancji. Dlatego też przed zastosowaniem PCA zalecana jest standaryzacja danych[3]. W dalszych rozważaniach brane będą pod uwagę dane zestandaryzowane.

Analiza składowych głównych posiada również interpretację graficzną (także u Webba [7]). Ponieważ wariancja to w istocie suma kwadratów odległości punktów danych od wartości średniej (po standaryzacji jest to środek układu współrzędnych), jej maksymalizacja wzdłuż pierwszego kierunku głównego jest równoznaczna z minimalizacją wariancji w pozostałych wymiarach.

Widać to wyraźnie rozpatrując wkład pojedynczego wektora $\mathbf{x}$ (w reprezentacji głównych składowych) do wariancji (patrz rys. 3.1):

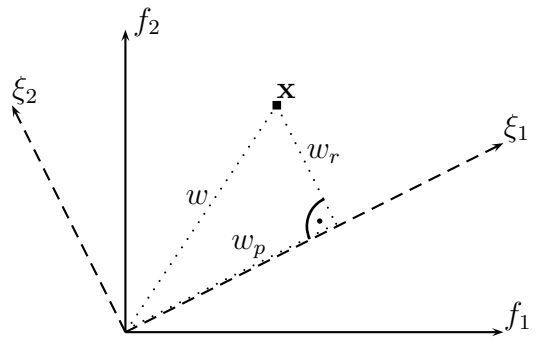
$$w^2 = (\bar{\mathbf{x}}-\mathbf{x})^2 = \sum_{f=1}^{d}(\bar{\mathbf{x}}[f]-\mathbf{x}[f])^2 = (\bar{\mathbf{x}}[1]-\mathbf{x}[1])^2+\sum_{f=2}^{d}(\bar{\mathbf{x}}[f]-\mathbf{x}[f])^2 = w_p^2+w_r^2$$

w_p^2 jest wkładem wektora $\mathbf{x}$ uwzględniającym (jedynie) pierwszy kierunek główny. Całkowita wariancja to:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n} w_i^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} w_{i,p}^2 + \sum_{i=1}^{n} w_{i,r}^2\right) = S_p^2 + S_r^2$$

[2]Znalezione rozwiązanie jest identyczne z przedstawionym w tej pracy.

[3]Standaryzacja oznacza transformację zbioru wartości $\{a_i\}$ według wzoru: $a_i \to \frac{a_i-\overline{a}}{S_a}$, gdzie $\overline{a}$ jest wartością średnią, a S_a — odchyleniem standardowym. W wyniku otrzymujemy zbiór $\{a'_i\}$, dla którego $\overline{a'} = 0$ oraz $S_{a'} = 1$.



Rysunek 3.1: Dwuwymiarowy wektor $\mathbf{x}$ w reprezentacji oryginalnych cech (f_1, f_2) oraz głównych składowych (ξ_1, ξ_2). Jego wkład do całkowitej wariancji to $w^2 = w_p^2 + w_r^2$.

Ale S^2 =const., ponieważ jest to wariancja całych danych, niezmiennicza względem transformacji PCA. Zatem maksymalizacja wariancji dla pierwszej głównej składowej (S_p^2) oznacza minimalizację S_r^2, czyli sumy kwadratów odległości punktów danych od osi wyznaczającej pierwszy kierunek główny.

Jest to zatem linia najlepszego dopasowania w sensie najmniejszych kwadratów. Uzyskane rozwiązanie jest jednak różne od otrzymanego z regresji liniowej. W tradycyjnej regresji wybiera się jedną ze zmiennych jako niezależną, a odległości do minimalizacji mierzy się wzdłuż drugiego kierunku.

Druga składowa główna odpowiada za maksimum wariancji w przestrzeni prostopadłej do pierwszej, zatem razem rozpinają one płaszczyznę najlepszego dopasowania tzn. o najmniejszej sumie kwadratów odległości liczonych w pozostałych wymiarach. Analogiczną konstrukcję przeprowadzić można dla kolejnych kierunków głównych.

3.2 Znajdowanie składowych głównych

Dane jest n wektorów w d-wymiarowej przestrzeni. Niech X będzie macierzą $d \times n$ zawierającą (w kolumnach) oryginalne wektory danych, a Y — te same wektory w nowej bazie. Szukana transformacja przeprowadzająca X w Y powinna zachowywać odległości i kąty, a jedyną dozwoloną operacją jest rotacja[4]. Takie transformacje noszą nazwę *ortogonalnych*; przejście ze starej

[4]Oraz odbicia lustrzane.

do nowej bazy można w takim przypadku zapisać przy pomocy *ortogonalnej*[5] macierzy Z o wymiarze $d \times d$:

$$Y = Z^T X, \tag{3.1}$$

gdzie kolumny Z są wektorami nowej bazy, a Z^T oznacza transpozycję macierzy. Jeśli macierz jest ortogonalna, to zachodzi:

$$Z^T Z = Z Z^T = \mathbb{1}, \tag{3.2}$$

gdzie $\mathbb{1}$— macierz jednostkowa[6]. Jest to równoznaczne z żądaniem *ortonormalności*[7] kolumn (i wierszy) Z. A zatem znalezienie ortonormalnej bazy wektorów o żądanych własnościach automatycznie definiuje transformację potrzebną do wyrażenia wektorów danych w nowej bazie.

Jak wspomniano wcześniej, składowe główne są nieskorelowane. Kowariancja dwóch cech[8] jest dana wzorem:

$$c_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} \mathbf{x}_k[i] \cdot \mathbf{x}_k[j] = \frac{1}{n-1} \sum_{k=1}^{n} X[i,k] \cdot X[j,k], \tag{3.3}$$

gdzie $X[i,k] = \mathbf{x}_k[i]$ oznacza i-tą składową k-tego wektora. W zapisie macierzowym:

$$C_X = \frac{1}{n-1} X X^T \tag{3.4}$$

C_X jest nazywana *macierzą kowariancji*. Element $C_X[i,j] = c_{ij}$ to kowariancja pomiędzy cechami i i j, a na diagonali znajdują się wartości wariancji w każdym z kierunków. $C_X[i,j] = 0$ dla cech nieskorelowanych, a zatem brak korelacji pomiędzy wszystkimi kombinacjami zmiennych oznacza, że C_X jest macierzą diagonalną.

Szukana transformacja dana przez równanie (3.1) ma więc za zadanie diagonalizację macierzy kowariancji. Dla nowych zmiennych zachodzi:

$$\begin{aligned} C_Y &= \frac{1}{n-1} Y Y^T = \frac{1}{n-1} (Z^T X)(Z^T X)^T = \frac{1}{n-1} Z^T X X^T (Z^T)^T = \\ &= Z^T \left(\frac{1}{n-1} X X^T \right) Z = Z^T C_X Z \end{aligned} \tag{3.5}$$

[5]Macierz ortogonalna to taka, dla której $Z^T = Z^{-1}$, gdzie Z^{-1} — macierz odwrotna.

[6]Macierz jednostkowa to macierz diagonalna z jedynkami na diagonali, $\mathbb{1}[i,j] = \delta_{ij}$.

[7]Ogólnie ortonormalny zbiór wektorów to taki, w którym dla dowolnych i i j zachodzi $\mathbf{z}_i \mathbf{z}_j = \delta_{ij}$.

[8]Przy założeniu, że wartość średnia każdej z cech jest zerem. Warunek ten jest spełniony po przeprowadzeniu standaryzacji danych (patrz str.25.).

Warto zaznaczyć, że nie wszystkie wartości diagonalne C_Y muszą być niezerowe. Znalezienie zer na diagonali oznacza, że wektory danych znajdują się w pewnej podprzestrzeni o wymiarze d' równym ilości niezerowych elementów diagonalnych C_Y. Wariancja dla ostatnich $d - d'$ kierunków głównych jest zerowa.

Można pokazać, że macierz Z daje się zbudować z *wektorów własnych* macierzy kowariancji dla starej bazy (C_X). Są to wektory $\mathbf{z}$ spełniające równanie:

$$C_X\mathbf{z} = \lambda\mathbf{z}, \tag{3.6}$$

gdzie λ jest skalarem nazywanym *wartością własną*. Każdemu wektorowi własnemu $\mathbf{z}$ odpowiada jedna λ, ale może się zdarzyć, że ma ona tę samą wartość dla różnych $\mathbf{z}$. Równanie (3.6) jest często przekształcane do postaci:

$$(C_X - \lambda\mathbb{1})\mathbf{z} = 0 \tag{3.7}$$

Jest to układ równań liniowych, gdzie niewiadomymi są składowe wektora $\mathbf{z}$. Warunkiem istnienia nietrywialnych ($\mathbf{z} \neq 0$) rozwiązań jest zerowanie się wyznacznika macierzy $(C_X - \lambda\mathbb{1})$. Wyrażenie $f(\lambda) = \det(C_X - \lambda\mathbb{1})$ jest w istocie wielomianem zmiennej λ, często nazywanym *wielomianem charakterystycznym*. Pozwala to wyznaczyć, przynajmniej w teorii, wartości własne macierzy C_X jako miejsca zerowe tego wielomianu. Dla każdej znalezionej wartości λ odpowiadający jej wektor własny znaleźć można z układu równań (3.7). Warto zauważyć, że wektor $\mathbf{z}$ spełniający równanie (3.6) można dowolnie skalować (czyli dla dowolnej liczby α wektor $\alpha\mathbf{z}$ też jest wektorem własnym); fakt ten jest wykorzystywany do normalizacji[9] $\mathbf{z}$.

C_X jest macierzą symetryczną[10] o wartościach rzeczywistych, a zatem wszystkie jej wektory i wartości własne są rzeczywiste. Co więcej, wektory własne odpowiadające różnym wartościom własnym są ortogonalne. Jeśli chodzi o wektory własne dla tej samej wartości własnej[11], to rozpinają one pewną podprzestrzeń, co nie stoi na przeszkodzie, żeby je zortogonalizować (np. metodą Grama-Schmidta). W ten sposób zawsze można znaleźć bazę ortonormalnych wektorów własnych $\mathbf{z}$ macierzy C_X. Jeżeli z takich wektorów utworzona zostanie macierz Z ($\mathbf{z}$ są kolumnami), to można napisać macie-

[9]Dobierane jest $\alpha = (\mathbf{z}^T\mathbf{z})^{-1/2}$, co daje $|\alpha\mathbf{z}| = \sqrt{(\alpha\mathbf{z})^T(\alpha\mathbf{z})} = 1$, czyli wektor $\alpha\mathbf{z}$ ma długość 1.

[10]Patrz wzór (3.3). Zamiana indeksów i i j prowadzi do $c_{ij} = c_{ji}$.

[11]Taką wartość własną nazywa się *zdegenerowaną*.

rzowy odpowiednik równania (3.6):

$$C_X Z = Z\Lambda, \tag{3.8}$$

gdzie Λ jest macierzą diagonalną, a $\Lambda[i,i] \overset{ozn.}{=} \lambda_i$ to wartość własna odpowiadająca wektorowi własnemu stojącemu w i-tej kolumnie macierzy Z.

Wracając do macierzy kowariancji dla składowych głównych, jeżeli za macierz Z przyjąć ortonormalną bazę wektorów własnych C_X, to korzystając z równania (3.5):

$$C_Y \overset{(3.5)}{=} Z^T C_X Z \overset{(3.8)}{=} Z^T Z\Lambda \overset{(3.2)}{=} \Lambda \tag{3.9}$$

Wynika z tego, że budując macierz Z z wektorów własnych macierzy kowariancji C_X w wyniku transformacji (3.1) dane zapisane zostaną w nowych zmiennych, dla których C_Y jest macierzą diagonalną. Kierunki główne są zatem nieskorelowane, tak jak wymagała tego pierwsza z własności PCA (str. 24.). Dodatkowo wariancję danych w nowej bazie określają wartości diagonalne Λ. Przypomnijmy, że są to wartości własne C_X. Ponieważ macierz ta jest dodatnio półokreślona[12], są one nieujemne, tak jak jest wymagane od wariancji. Wartości te odpowiadają konkretnym wektorom własnym C_X, czyli kolumnom macierzy transformacji Z. Sortując macierz Λ i odpowiednio kolumny Z według malejących λ_i otrzymuje się uporządkowanie będące treścią drugiej z własności wymienionych na str. 24.

3.2.1 Znajdowanie wektorów własnych

Jak pokazano wyżej, zarówno znalezienie głównych składowych, jak i transformacji danych ze starych do nowych współrzędnych sprowadza się do znalezienia wektorów i wartości własnych macierzy kowariancji C_X. W tym celu wygodnie jest użyć rozkładu ze względu na wartości osobliwe (ang. *Singular Value Decomposition* = SVD). Dla dowolnej macierzy A rozkład ten wygląda następująco:

$$A = U\Sigma V^T, \tag{3.10}$$

gdzie U i V są macierzami o ortonormalnych kolumnach (patrz str. 27.), a Σ — macierzą diagonalną. Kolumny U to lewe-, a kolumny V — prawe

[12]Macierz A jest dodatnio półokreślona, jeżeli dla dowolnego wektora $\mathbf{x} \neq 0$ zachodzi $\mathbf{x}^T A\mathbf{x} \geqslant 0$. Macierze w postaci $A = BB^T$ są zawsze dodatnio półokreślone, ponieważ $\mathbf{x}^T(BB^T)\mathbf{x} = (B^T\mathbf{x})^T(B^T\mathbf{x}) = \mathbf{y}^T\mathbf{y} = |\mathbf{y}|^2 \geqslant 0$ z definicji. Macierz C_X jest postaci BB^T (patrz równanie 3.4).

wektory osobliwe (ang. *left/right singular vectors*). Natomiast elementy diagonalne $\Sigma[i,i] \stackrel{ozn.}{=} \sigma_i$ noszą nazwę wartości osobliwych (ang. *singular values*). Wszystkie σ_i są nieujemne. Rezultatem mnożenia A przez A^T jest:

$$AA^T = U\Sigma V^T(U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T = U(\Sigma\Sigma^T)U^T \tag{3.11}$$

$$(AA^T)U = U(\Sigma\Sigma^T) \tag{3.12}$$

Porównując ostatnie równanie z (3.8) widać, że lewe wektory osobliwe macierzy A są wektorami własnymi AA^T, a wartości osobliwe A — pierwiastkami wartości własnych AA^T.

Niech[13] A będzie macierzą danych ($A \to X$), a U — macierzą transformacji ($U \to Z$). Dzielenie (3.12) przez $(n-1)$ daje:

$$\left(\frac{1}{n-1}XX^T\right)Z = Z\left(\frac{1}{n-1}\Sigma\Sigma^T\right) \tag{3.13}$$

$$C_X Z = Z\left(\frac{1}{n-1}\Sigma\Sigma^T\right), \tag{3.14}$$

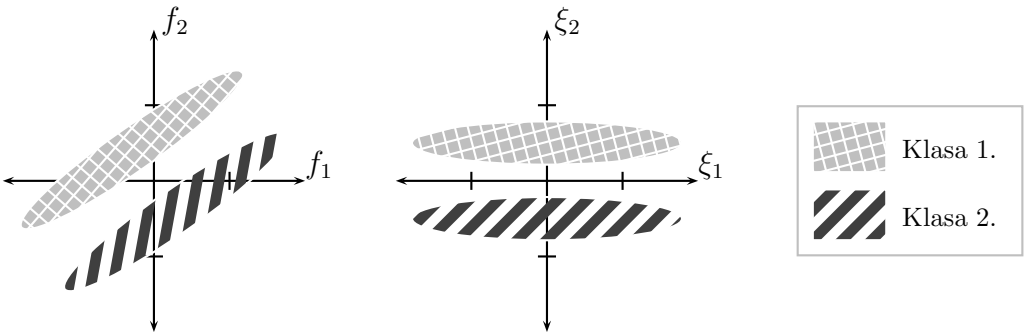
z czego wynika, że $\Lambda = \frac{1}{n-1}\Sigma\Sigma^T$, czyli $\lambda_i = \frac{\sigma_i^2}{n-1}$. Otrzymany rezultat oznacza, że przeprowadzając rozkład macierzy danych X ze względu na wartości osobliwe dostaje się od razu macierz wektorów własnych C_X (jest to macierz U) i można łatwo wyznaczyć jej wartości własne λ_i przy pomocy wartości osobliwych σ_i. Nie jest konieczne wyznaczanie samej macierzy kowariancji C_X.

3.3 Selekcja rankingowa na bazie PCA

Przestrzeń kierunków głównych gwarantuje co prawda brak korelacji pomiędzy atrybutami oraz ich uszeregowanie zgodnie z malejącą wariancją w danych, ale z punktu widzenia selekcji cech posiada dwie istotne wady. Po pierwsze, składowe odpowiadające za dużą wariancję w danych niekoniecznie nadają się dobrze do dyskryminacji (patrz rys 3.2). Po drugie, nowe zmienne są, jak pokazano dokładnie poniżej, pewną liniową kombinacją starych. Trudno zatem przypisać im jakieś konkretne znaczenie i nawet jeśli dają dobre wyniki klasyfikacji, nie wnosi to nic do rozumienia problemu.

Pierwszej z wymienionych wad trudno zaradzić; w zależności od rozkładu danych PCA jest bardziej lub mniej użyteczną techniką. Jeżeli chodzi o

[13] Jest to jedynie zmiana oznaczeń.



Rysunek 3.2: Przykład rozkładu danych z dobrze rodzielonymi klasami (w oryginalnej przestrzeni (f_1, f_2), po lewej). Rzutowanie na pierwszą główną składową (ξ_1) zupełnie usuwa separację; zostaje ona zachowana dla drugiej głównej składowej (ξ_2).

trudność interpretacji głównych składowych, na podstawie PCA można wyznaczyć współczynniki oceniające oryginalne cechy i przy ich pomocy przeprowadzić selekcję rankingową (patrz rozdziały 1.2.1. i 2.).

Przyjmując, że składowe odpowiadające za większą wariancję są „lepsze", należy również faworyzować cechy, które wnoszą największy wkład do pewnej ograniczonej liczby pierwszych kierunków głównych.

Wracając zatem do równania transformacji (3.1), tym razem dla pojedynczych wektorów $\mathbf{x}$ i $\mathbf{y}$, można napisać:

$$\mathbf{y} = Z^T\mathbf{x} = \begin{pmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_d^T \end{pmatrix} \mathbf{x}, \tag{3.15}$$

gdzie $\mathbf{z}_i$ jest wektorem wyznaczającym i-ty kierunek główny.

Widać stąd, że $\mathbf{y}[i] = \mathbf{z}_i^T\mathbf{x}$, czyli i-ta składowa wektora $\mathbf{y}$ jest rzutem $\mathbf{x}$ na kierunek $\mathbf{z}_i$, tak jak można się było tego spodziewać. Jeśli ograniczyć wymiarowość danych do pierwszej głównej składowej $\mathbf{z}_1$, prowadzi to do:

$$\mathbf{y} = \mathbf{z}_1^T\mathbf{x} = \sum_{j=1}^{d} \mathbf{z}_1[j]\mathbf{x}[j] \tag{3.16}$$

Powyższy wzór można rozumieć w ten sposób, że wektor $\mathbf{y}$ (tu jednowymiarowy) jest kombinacją liniową składowych odpowiadającego mu wektora

$\mathbf{x}$, a składowe wektora pierwszego kierunku głównego $\mathbf{z}_1$ pełnią rolę współczynników tej kombinacji. Znaczy to, że j-ta składowa wektora $\mathbf{z}_1$ określa, jaki wkład wnosi j-ta cecha z oryginalnej przestrzeni do pierwszego kierunku głównego.

Rozpatrując jedynie pierwszy kierunek główny widać, że wartościami indeksu, według którego tworzony jest ranking mogą być po prostu składowe $\mathbf{z}_1$. Ale pozostawienie k głównych składowych daje dla każdej cechy k wartości (i-te składowe wektorów $\mathbf{z}_1$,...,$\mathbf{z}_k$), które oceniają jej wkład do 1,2,...,k-tej głównej składowej. Maksimum tych wartości mówi, jaki jest największy wkład danej cechy do jednej spośród k pierwszych PC. Suma z kolei pozwala oszacować całkowity wpływ rozpatrywanej cechy na wszystkie k głównych składowych.

Schemat selekcji z użyciem PCA ma zatem dwa parametry (k i liczbę wybieranych cech r) oraz dwa rodzaje indeksu (suma i maksimum) i wygląda następująco:

1. Rozkład SVD standaryzowanej macierzy danych $X = U\Sigma V^T$.
2. Wyznaczenie macierzy Z poprzez sortowanie kolumn U względem odpowiadających im wartości $\lambda_i = \frac{\sigma_i^2}{n-1}$.
3. Ograniczenie liczby kolumn Z do k; powstała macierz to Z'.
4. Wyznaczenie sumy/maksimum elementów dla każdego wiersza Z'.
5. Utworzenie rankingu cech i pozostawienie r najlepszych.

Selekcja rankingowa na bazie PCA działająca według powyższego schematu została w ramach tej pracy zaimplementowana jako moduł dla pakietu analizy danych GhostMiner. Działający model został następnie przetestowany na zbiorach z konkursu NIPS 2003 Feature Selection Challenge (patrz rozdz. 5.1.). Najciekawsze wyniki znaleźć można w rozdziale 5.3.

Rozdział 4

Poprawność numeryczna naiwnego klasyfikatora Bayesowskiego

Na potrzeby tego rozdziału, w celu zapewnienia lepszej czytelności wzorów, składowe wektorów oznaczane będą według: $\mathbf{x}[f] \rightarrow x_f$.

4.1 Naiwny klasyfikator Bayesowski

Prawdopodobieństwo, że wektor $\mathbf{x}$ należy do klasy $c \in C$, jeżeli wiadomo, jakie wartości przyjmują jego poszczególne atrybuty można zapisać jako $P(c|\mathbf{x})$. Jest to prawdopodobieństwo *a posteriori* w odróżnieniu od prawdopodobieństwa przynależności dowolnego przypadku do c, które jest określane jako *a priori* i oznaczane $P(c)$. Optymalna decyzja klasyfikacyjna to przydzielenie $\mathbf{x}$ do najbardziej prawdopodobnej (= np) klasy:

$$c_{\text{np}} = \arg\max_{c \in C} P(c|\mathbf{x}) \tag{4.1}$$

Korzystając z wzoru Bayesa:

$$c_{\text{np}} = \arg\max_{c \in C} \frac{P(\mathbf{x}|c)P(c)}{P(\mathbf{x})} = \arg\max_{c \in C} P(\mathbf{x}|c)P(c) \tag{4.2}$$

Model działający w oparciu o wzór (4.2) — przy założeniu, że występujące w nim prawdopodobieństwa są znane lub można je oszacować — nosi nazwę optymalnego klasyfikatora Bayesowskiego.

Dla cech niezależnych zachodzi:

$$P(\mathbf{x}|c) = P(x_1 \wedge x_2 \wedge \cdots x_d|c) = \prod_{i=1}^{d} P(x_i|c) \tag{4.3}$$

Naiwny klasyfikator Bayesowski (= NB) stosuje formułę (4.3) — jako przybliżenie — dla wszystkich zestawów cech, bez względu na istnienie lub brak zależności. Prowadzi to do wyboru klasy:

$$c_{\mathrm{NB}} = \arg\max_{c \in C} P(c) \prod_{i=1}^{d} P(x_i|d) \tag{4.4}$$

4.2 Dokładność numeryczna algorytmu

Prawdopodobieństwa ze wzoru (4.4) są zazwyczaj, dla cech numerycznych, przybliżane wartościami funkcji Gaussa:

$$P(x_i|c) = N(x_i; \mu_{i,c}, \sigma_{i,c}) = \frac{1}{\sqrt{2\pi}\sigma_{i,c}} \exp\left(-\frac{(x_i - \mu_{i,c})^2}{2\sigma_{i,c}^2}\right), \tag{4.5}$$

gdzie $\mu_{i,c}$ oraz $\sigma_{i,c}$ są odpowiednio: wartością średnią i odchyleniem standardowym dla cechy i wektorów z klasy c. Takie przybliżenie odpowiada przyjęciu założenia, że wartości atrybutu i podlegają rozkładowi normalnemu.

Jeśli pominąć prawdopodobieństwo aprioryczne $P(c)$, do wyznaczenia pozostaje iloczyn wartości $P(x_i|c)$ określonych jak w równaniu (4.5). Jego wartość oczekiwana jest dana przez:

$$E\left(\prod_{i=1}^{d} P(x_i|c)\right) = \int_{\mathbb{R}^d} \prod_{i=1}^{d} P(x_i|c) \cdot \rho(\mathbf{x}) \mathrm{d}\mathbf{x}, \tag{4.6}$$

gdzie ρ jest gęstością rozkładu prawdopodobieństwa wektorów danych z klasy c. Ale jeśli spełniony jest warunek niezależności cech oraz założenie o normalności ich rozkładów, to:

$$\rho(\mathbf{x}) = \prod_{j=1}^{d} N(x_j; \mu_{j,c}, \sigma_{j,c}), \tag{4.7}$$

co w efekcie daje:

$$E\left(\prod_{i=1}^{d} P(x_i|c)\right) =$$

$$= \int_{\mathbb{R}^d} \left[\prod_{i=1}^{d} N(x_i; \mu_{i,c}, \sigma_{i,c})\right] \left[\prod_{j=1}^{d} N(x_j; \mu_{j,c}, \sigma_{j,c})\right] \mathrm{d}x_1 \cdots \mathrm{d}x_d =$$

$$= \prod_{i=1}^{d} \int_{\mathbb{R}} \left[N(x_i; \mu_{i,c}, \sigma_{i,c})\right]^2 \mathrm{d}x_i = \prod_{i=1}^{d} \frac{1}{4\sqrt{\pi}\sigma_{i,c}} \tag{4.8}$$

Jeśli dla uproszczenia założyć jednakowe rozkłady we wszystkich wymiarach (czyli $\sigma_{i,c} = \sigma_c$ dla $i = 1, \ldots, d$), to z (4.8) wynika, że:

$$E\left(\prod_{i=1}^{d} P(x_i|c)\right) = \left(4\sqrt{\pi}\sigma_c\right)^{-d} \tag{4.9}$$

Szacowanie prawdopodobieństwa $P(c|\mathbf{x})$ w naiwnym klasyfikatorze Bayesowskim wymaga wyznaczenia iloczynu danego przez (4.4), którego wartość oczekiwana jest, pod warunkiem spełnienia stosowanych założeń, dana równaniem (4.9). Jeśli w trakcie tej operacji nastąpi przekroczenie zakresu używanego typu zmiennoprzecinkowego, to jej wynikiem będzie zero. Oznacza to, że informacja o prawdopodobieństwach warunkowych $P(x_i|c)$ zostanie utracona, a naiwny klasyfikator Bayesowski będzie w najlepszym wypadku działać jak klasyfikator większościowy.

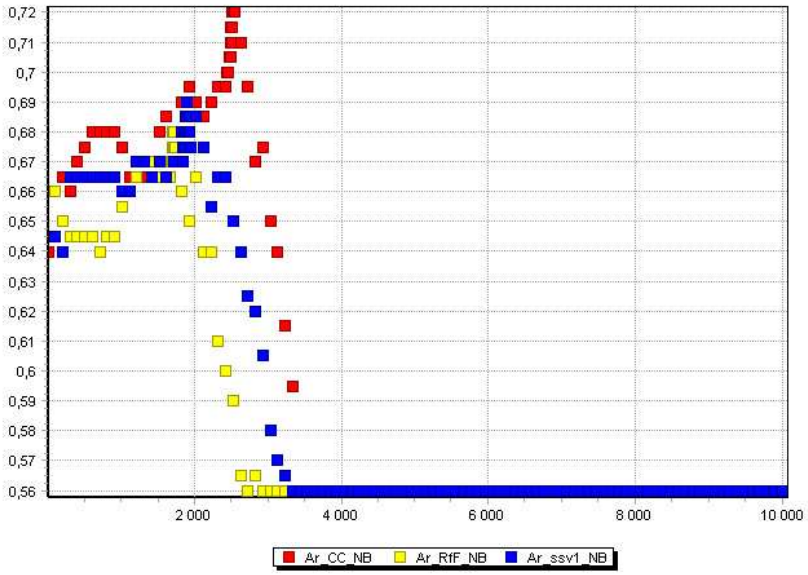
Precyzja	Bity	Najmniejsza wartość dodatnia
pojedyncza	32	$2^{-149} \approx 10^{-44.85}$
podwójna	64	$2^{-1074} \approx 10^{-323.3}$

Tabela 4.1: Najmniejsze liczby (co do wartości bezwzględnej) reprezentowalne w pojedynczej i podwójnej precyzji (wartości zdenormalizowane).

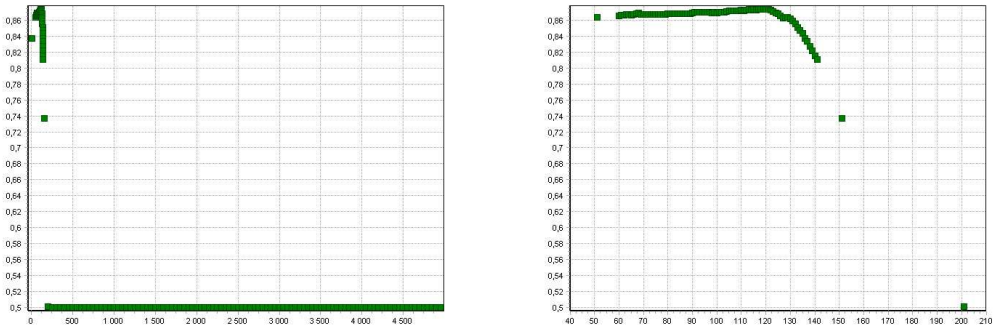
Jakie wartości d wystarczą, aby tak się stało? Tabela 4.1 zawiera krótkie podsumowanie własności najpopularniejszych typów zmiennoprzecinkowych (zgodnie ze standardem IEEE 754 [8]). Zakładając standaryzację danych, rozkłady wartości cech mają postać $N(x_i; 0, 1)$, ale odchylenia w ramach klas są zazwyczaj mniejsze niż 1. Jeśli przyjąć $\sigma_c = \frac{1}{4}$, to przekroczenia zakresu dla pojedynczej precyzji wystarczy $d = 181$, a dla podwójnej — $d = 1301$. Wartości te stanowią orientacyjną granicę przydatności naiwnego klasyfikatora Bayesowskiego. Jeżeli wymiarowość przestrzeni cech jest zbliżona do 180 (lub 1300, w zależności od używanej precyzji), to obserwowana stopniowa degradacja wyników tego modelu przy dodawaniu kolejnych atrybutów nie

musi wynikać z ich nieprzydatności, ale z zerowania $P(c|\mathbf{x})$ będącego skutkiem mnożenia szeregu małych wartości.

Rysunki 4.1 i 4.2 pokazują przykłady utraty dokładności naiwnego klasyfikatora Bayesowskiego dla różnych metod selekcji, na różnych zbiorach danych.



Rysunek 4.1: Utrata dokładności naiwnego klasyfikatora Bayesowskiego korztystającego z różnych metod selekcji na zbiorze Arcene (patrz rozdz. 5.1).



Rysunek 4.2: Utrata dokładności naiwnego klasyfikatora Bayesowskiego na zbiorze Gisette (patrz rozdz. 5.1). Po prawej stronie w powiększeniu obszar bliski maksymalnej osiągniętej dokładności.

Rozdział 5

Analiza danych z konkursu NIPS 2003 Feature Selection Challenge

5.1 Konkurs

Konkurs selekcji cech NIPS[1] 2003 Feature Selection Challenge był oparty na pięciu zbiorach danych: Arcene, Dexter, Dorothea, Gisette i Madelon. Problemy klasyfikacyjne były dwuklasowe i wysokowymiarowe. Podstawowe własności zbiorów zostały podsumowane w tabeli 5.1. Każdy z nich został podzielony na trzy części: treningową, walidacyjną i testową. W fazie finałowej oraz później, po zakończeniu konkursu dostępne były etykiety klas dla punktów wchodzących w skład dwóch pierwszych części. Przewidywane wyniki klasyfikacji na ostatniej (testowej) części można było składać na stronie konkursu (`http://www.nipsfsc.ecs.soton.ac.uk/`). Były one oceniane przy wykorzystaniu publicznie niedostępnych etykiet klas dla zbiorów testowych. Niestety, w czasie pisania tej pracy składanie wyników do oceny było niemożliwe z uwagi na przeprowadzane zmiany w konfiguracji serwerów.

5.2 Analiza danych

W celu znalezienia skutecznych modeli dla danych konkursowych dla każdego zbioru przeprowadzono testy szeregu par: selektor i klasyfikator. Spraw-

[1]Neural Information Processing Systems, `http://www.nips.cc/`

Zbiór	Liczba cech	Stosunek klas	Liczba pkt treningowych	Liczba pkt validacyjnych	Liczba pkt testowych
Arcene	10000	56/44	100	100	700
Dexter	20000	50/50	300	300	2000
Dorothea	100000	90/10	800	350	800
Gisette	5000	50/50	6000	1000	6500
Madelon	500	50/50	2000	600	1800

Tabela 5.1: Liczba cech, orientacyjny stosunek klas i liczebności poszczególnych części zbiorów konkursu NIPS 2003.

dzanymi algorytmami były metody selekcji opisane w rozdziale 2. oraz klasyfikatory: SVM (dwie wersje), naiwny klasyfikator Bayesowski (patrz rozdz. 4), kNN oraz drzewa SSV. Dla najlepszych metod klasyfikacji testowana była również selekcja w oparciu o PCA (patrz rozdz. 3.3.). Testy polegały na szukaniu optymalnej, w danym kontekście, liczby cech wybieranej przez selektor.

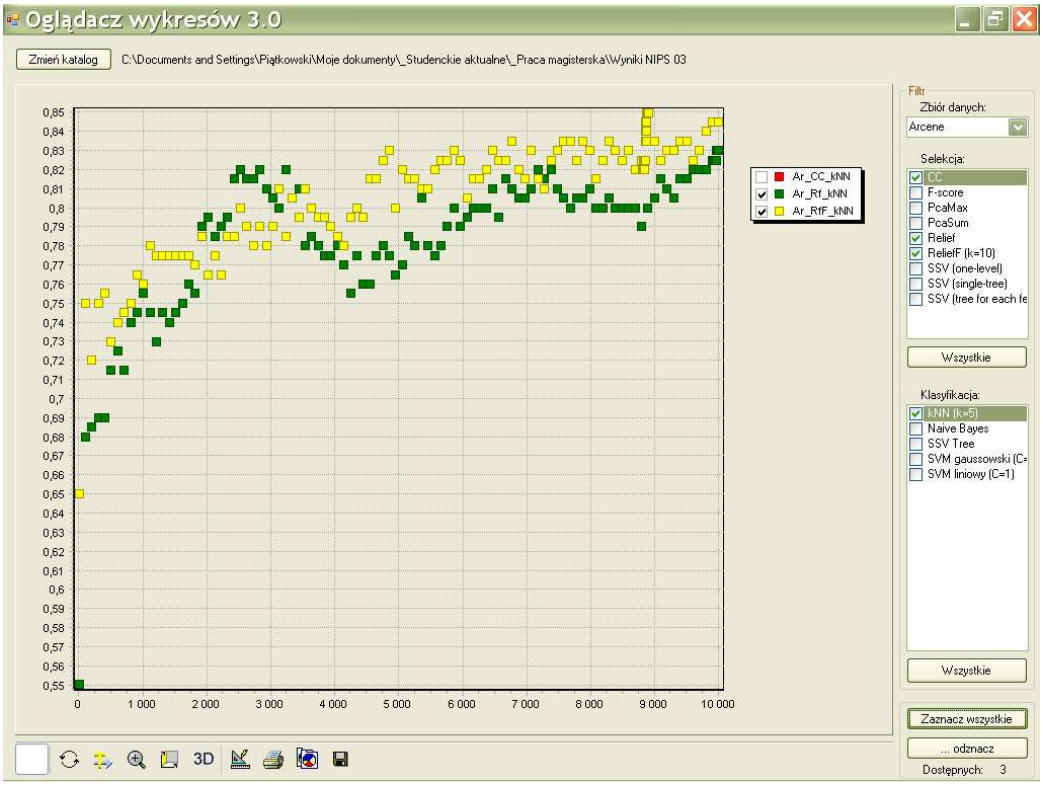
Dla najbardziej obiecujących modeli zostało przeprowadzone lokalne przeszukiwanie parametrów zarówno selektorów, jak i klasyfikatorów. Uzyskane w ten sposób konfiguracje są opisane w rozdziale 5.3 oraz zebrane w tabeli 5.3.

5.2.1 „Oglądacz wykresów"

Wynikiem opisanych wyżej testów było prawie 200 tekstowych plików wynikowych zawierających, dla wybranej konfiguracji, w kolumnach: liczbę cech wybranych przez selektor i dokładność modelu oszacowaną przez krosswalidację. Oprócz tego zapisywane były (w postaci plików `.eps`) wykresy zawierające te same dane, jak w pliku tekstowym: dokładność klasyfikacji w funkcji liczby cech.

W celu porównywania wyników dla różnych metod powstał program *Oglądacz wykresów* (patrz rys. 5.1). Na podstawie nazw plików rozpoznaje on konfigurację modelu oraz zbiór danych dla jakiego prowadzone były obliczenia. Umożliwia to wyświetlanie wielu wyników na jednym wykresie. Jednocześnie istnieje możliwość dowolnego filtrowania dostępnych serii danych (ze względu na zbiór danych, selektory i klasyfikatory) i wyboru, które są widoczne. Na przykład, na rysunku 5.1 widać porównanie różnych algorytmów selekcji dla klasyfikatora kNN na zbiorze Arcene. Sprawdzane są metody: współczyn-

nika korelacji (nie jest aktualnie wyświetlana) i Relief (2 wersje). Uzyskane wykresy można zapisywać do pliku, a interesujące obszary — powiększać.



Rysunek 5.1: Oglądacz wykresów

Program został napisany w języku C♯. Wykorzystano komponenty do obsługi i rysowania wykresów `TeeChart` firmy Steema.

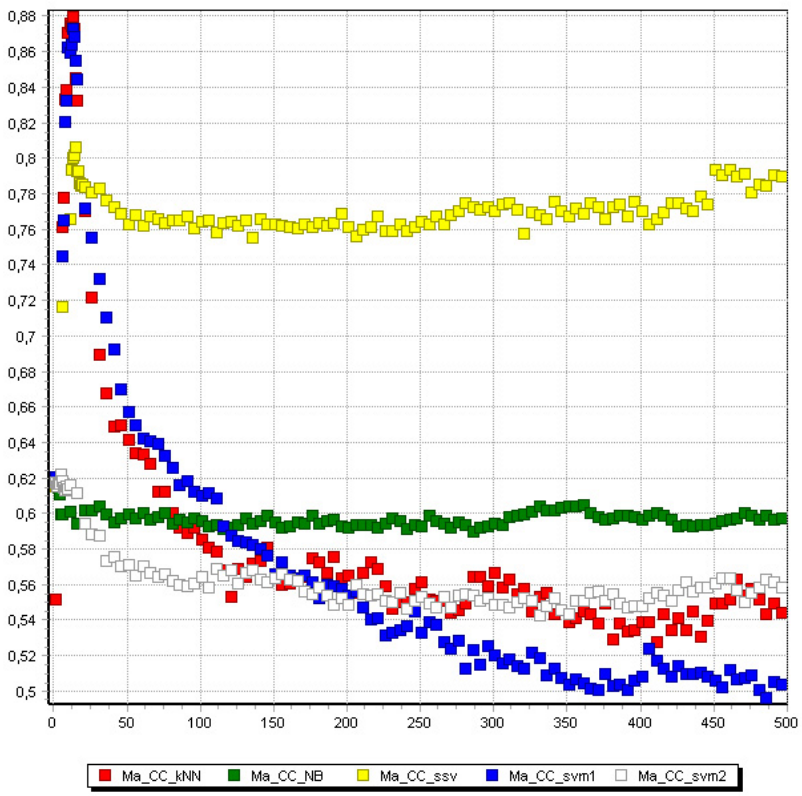
5.2.2 Wybrane przypadki

Zamieszczone w tym rozdziale wykresy mają automatycznie generowane legendy. Każda seria danych ma nazwę składającą się z trzech części oddzielonych znakiem podkreślenia (_). Są to skróty oznaczające: zbiór danych, selektor i klasyfikator, np. `Ma_CC_kNN` to dane opisujące dokładność klasyfikatora kNN po selekcji metodą współczynnika korelacji na zbiorze Madelon. SVM1 to w tej konwencji wersja gaussowska, a SVM2 — liniowa. Na osi poziomej znajduje się liczba cech wybrana przez selekcję, a na pionowej —

dokładność modelu. Obszary bliskie maksimum są spróbkowane dokładniej od reszty.

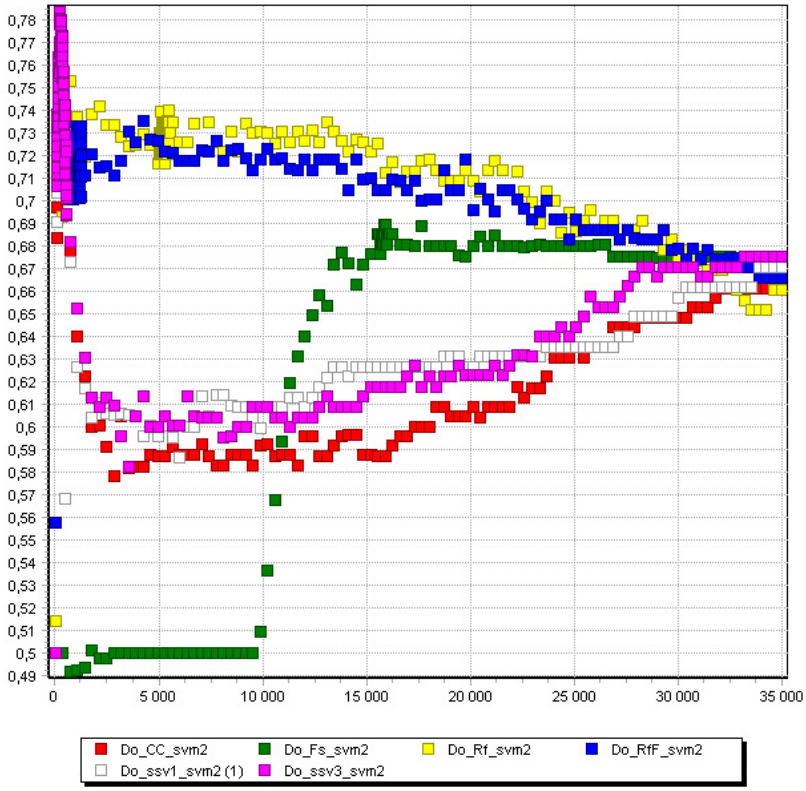
Analizując tak dużą liczbę złożonych modeli napotkać można ciekawe sytuacje wskazujące na pewne nieoczywiste własności.

Jeśli na przykład spojrzeć na rysunek 5.2, widać wyraźnie, że metody osiągające dokładność znacząco lepszą od innych na całym zbiorze danych (SSV,NB) nie poprawiają swoich wyników dzięki selekcji. W tym przypadku wydatnie pomaga ona SVM i kNN, które nie wypadają najlepiej działając na wszystkich cechach, ale dla niewielkiego ich podzbioru dystansują wszystkie inne modele. Oznacza to, że dokładność metody na całych danych nie musi świadczyć o jej potencjalnych możliwościach w wypadku zastosowania selekcji.



Rysunek 5.2: Zbiór Madelon. Selekcja cech poprawia dokładność jedynie niektórych modeli.

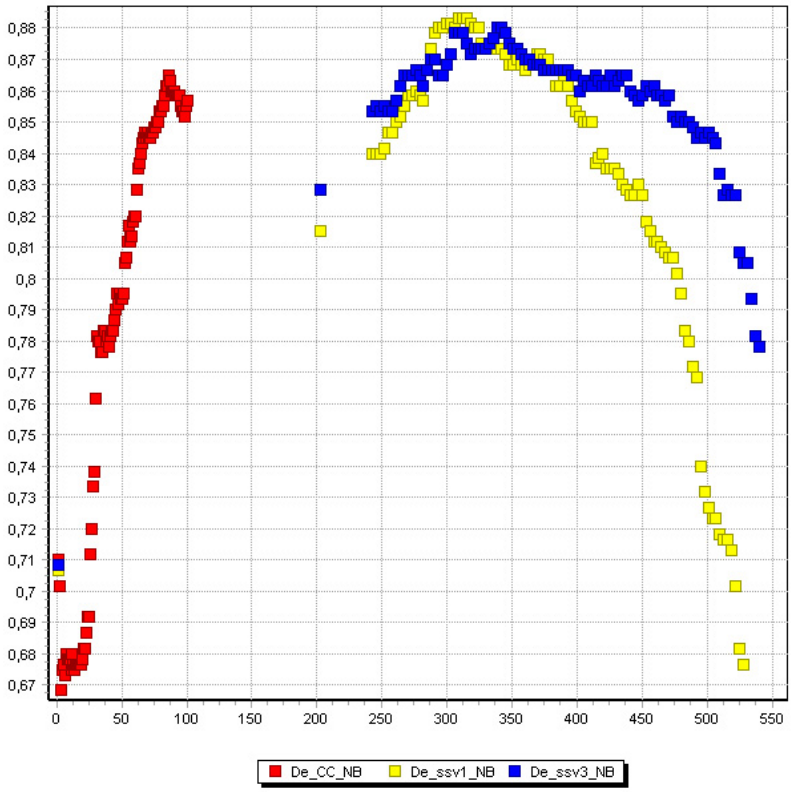
Zazwyczaj dla tego samego zbioru danych i klasyfikatora różne metody selekcji zachowują się podobnie — w szczególności pomagają albo nie. Ale rysunek 5.3 pokazuje, że nie zawsze tak jest. Widać na nim trzy grupy metod. Pierwsza to algorytmy z rodziny Relief. Poprawiają one nieco wyniki klasyfikatora i są bardzo stabilne (wolne zmiany dla różnej liczby cech). Grupa druga to CC i dwa warianty selekcji SSV. Pozwalają one klasyfikatorowi osiągnąć bardzo dobre wyniki, ale maksimum jest bardzo wąskie — trzeba dobrać właściwą liczbę cech, co może wymagać lokalnej optymalizacji tego parametru. Trzecią grupę tworzy selektor F-score, który w tym przypadku działa rażąco źle, tak jakby wybierał najpierw najmniej przydatne atrybuty.



Rysunek 5.3: Zbiór Dorothea i liniowy SVM. Różne skutki użycia selekcji w tym samym kontekście.

Z kolei na rysunku 5.4 widać, że dla różnych metod selekcji maksimum dokładności, nawet jeśli jest na podobnym poziomie, nie musi być dla tej

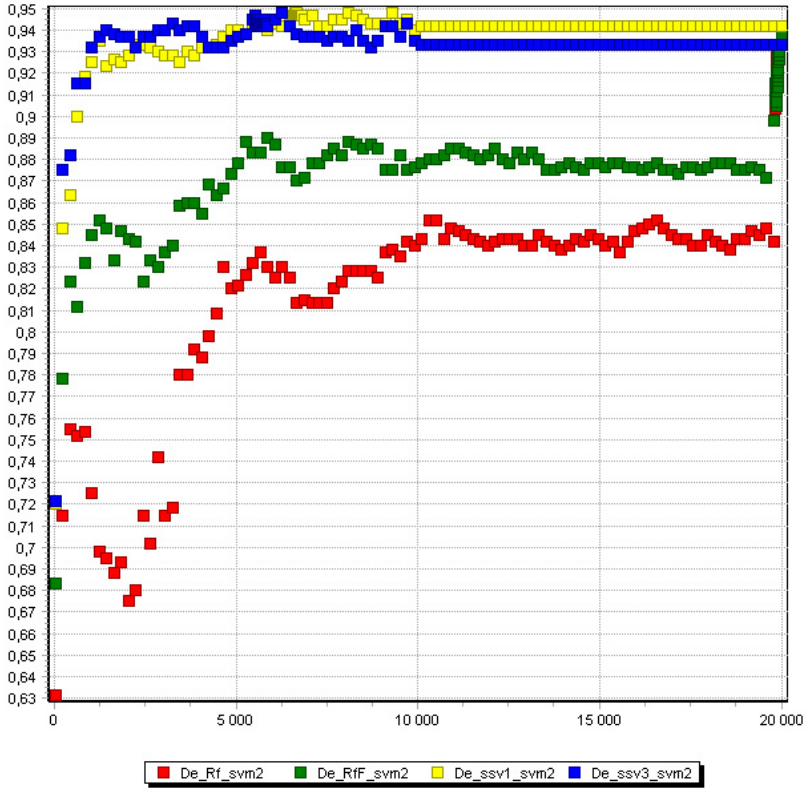
samej liczby cech.



Rysunek 5.4: Dexter i naiwny klasyfikator Bayesowski. Położenie maksimów dokładności dla różnych selektorów (powiększenie — zbiór ma 20000 cech).

Wykres 5.5 pokazuje bardzo ciekawy aspekt dotyczący metod Relief i ReliefF. Zbiór Dexter zawiera prawie 9000 cech o zerowej wariancji (wszystkich: 20000). Nie niosą one żadnej informacji i są bezużyteczne dla klasyfikatorów. Większość metod rankingowych ocenia takie atrybuty poprawnie i umieszcza je na końcu rankingu. Przykładem mogą być najlepsze w tym kontekście algorytmy SSV (one-level i tree-for-each-feature). Dokładność modeli, w których skład wchodzą, stabilizuje się w okolicy 11000 cech, a dodawanie nowych atrybutów nie ma na nią najmniejszego wpływu.

Ale zachowanie tego samego klasyfikatora (liniowa wersja SVM) jest zupełnie inne, jeśli korzysta on z selekcji Relief (ReliefF). W całym zakresie zmienności parametru (chodzi o liczę wybieranych cech) daje się zaobserwo-



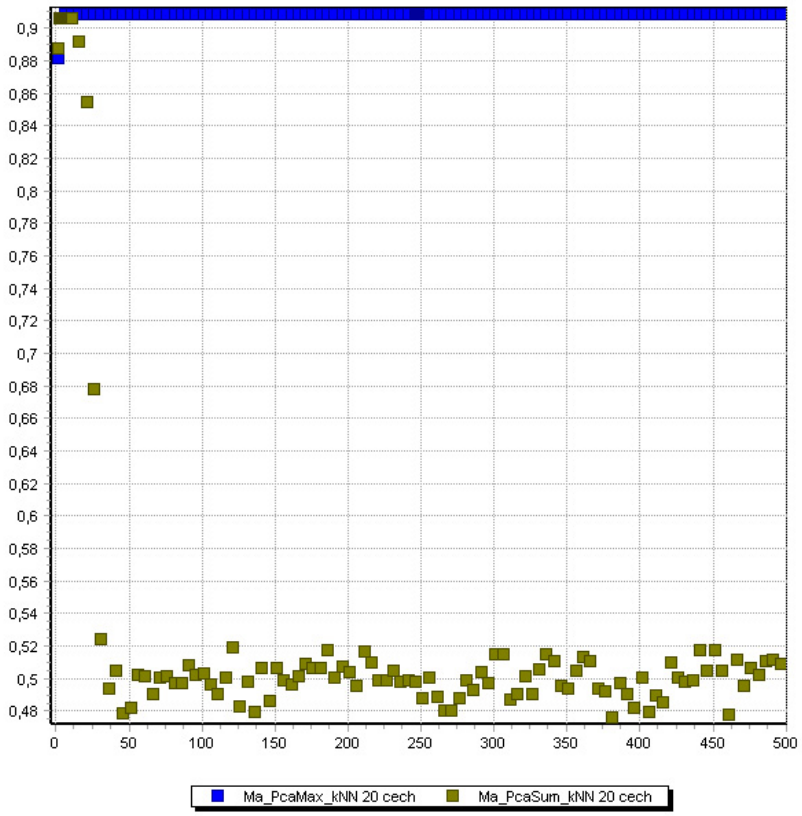
Rysunek 5.5: Zbiór Dexter i liniowy SVM. Wpływ atrybutów o zerowej wariancji na metody z rodziny Relief. Dla porównania najlepsze algorytmy.

wać wariancja dokładności. Wyniki — a należy tu raz jeszcze podkreślić, że chodzi o ten sam klasyfikator — są wyraźnie gorsze, niż dla najlepszych metod selekcji. Dopiero dla bardzo dużych podzbiorów (prawie wszystkie cechy) dokładność rośnie do spodziewanego poziomu. Dowodzi to, że algorytmy Relief ustawiają, w pewnych warunkach, atrybuty o zerowej wariancji wyżej w rankingu, niż zmienne o niezerowej wariancji. Co więcej, jak widać na rozpatrywanym przykładzie, są to cechy wartościowe, pomocne w klasyfikacji.

Można spekulować, kiedy tak się dzieje. Patrząc na wzory (2.8) i (2.9) nietrudno sprawdzić, że dla cech o zerowej wariancji wartość indeksu Relief (i ReliefF) będzie wynosiła 0. Czy może być ujemna? Niech dla każdej występującej wartości cechy f istnieją dokładnie 2 wektory posiadające taką wartość i niech należą one do różnych klas. Wtedy dodatnie składniki ze

wzorów (2.8) i (2.9) się zerują, a ujemne — nie. To pokazuje, że również dla atrybutów wykazujących się dużym przemieszaniem klas (podobne wartości średnie i odpowiednio duże wariancje) wartości indeksów Relief i ReliefF mogą być ujemne. Prawdopodobnie sprzyja temu również stosunkowo mała liczba wektorów, niewystarczająca do wiarygodnej oceny rozkładu klas.

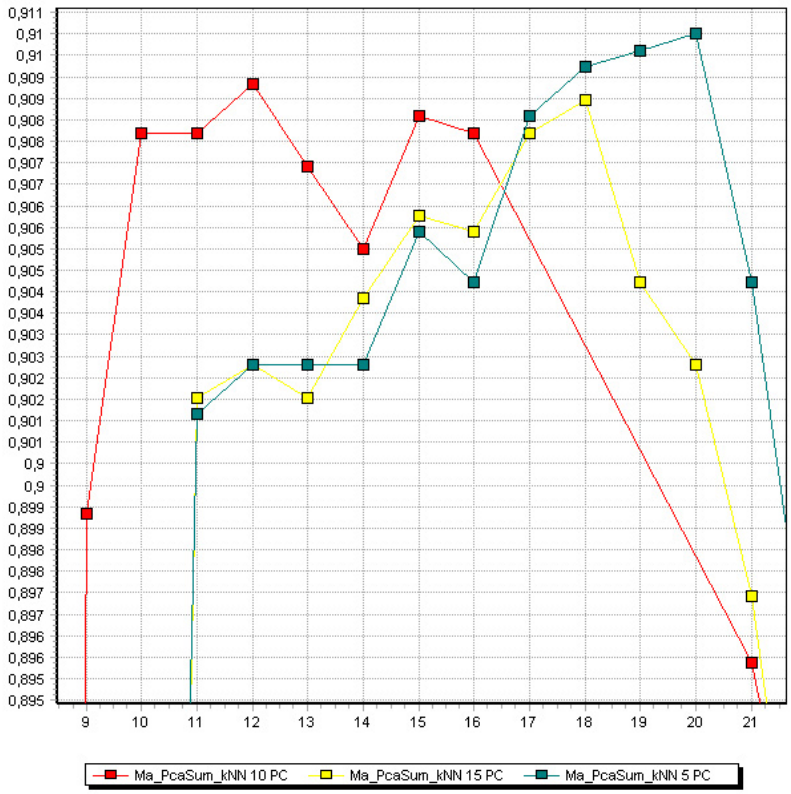
Wniosek jest taki, że przed zastosowaniem selekcji rankingowej Relief (ReliefF) należy koniecznie sprawdzić, czy zbiór nie zawiera atrybutów o zerowej wariancji i, jeśli trzeba, usunąć je.



Rysunek 5.6: Zbiór Madelon i kNN. Dokładność klasyfikacji na 20 cechach w funkcji liczby pierwszych głównych składowych rozpatrywanych przez selekcję PCA dla dwóch różnych kryteriów.

Ostatnia obserwacja dotyczy porównania kryteriów (suma i maksimum, patrz rozdz. 3.3) dla selekcji rankingowej PCA. Wykres 5.6 przedstawia do-

kładność tego selektora dla obu kryteriów w zależności od uwzględnianej liczby głównych składowych. Liczba wybieranych cech jest stała i wynosi 20. Jest to wartość bliska optymalnej. Zarówno suma, jak i maksimum działają bardzo dobrze dla niewielkiej liczby kierunków głównych. Jednak pierwsze z kryteriów traci przydatność przy zwiększaniu ich liczby, podczas gdy maksimum zachowuje się nad wyraz stabilnie.



Rysunek 5.7: Zbiór Madelon i kNN. Zachowanie selekcji PCA (kryterium: suma) w pobliżu maksimum dla różnej liczby pierwszych głównych składowych (5, 10 i 15). Punkty połączono w celu zapewnienia lepszej czytelności wykresu.

To jednak nie wszystko. Rysunek 5.7 przedstawia powiększony obszar bliski maksimum dokładności dla selekcji PCA, gdy kryterium jest suma. Zobaczyć na nim można różnice w osiąganiu maksimum dla różnej liczby rozpatrywanych kierunków głównych. Najlepszy podzbiór wybrany przez selekcję

korzystającą z 5 głównych składowych składa się z 20 cech. Jeśli zwiększyć ilość rozpatrywanych kierunków do 10, maksimum osiągane jest wcześniej, już dla 12 atrybutów. Ale dalsze zwiększanie liczby branych pod uwagę składowych głównych (do 15) ustawia maksimum z powrotem w okolicach 18 cech.

5.3 Wyniki

Podstawową miarą oceny modeli konkursowych jest błąd zbalansowany (ang. *Balanced Error Rate* = BER), czyli średnia błędów dla obu klas. Dodatkowo podane jest pole pod krzywą ROC (ang. *Area Under Curve* = AUC), procent użytych cech (ang. *Fraction of Features* = FF) oraz jaką ich część stanowiły atrybuty dodatkowe, wygenerowane przez organizatorów konkursu. Takie zmienne (określane jako *probes*, stąd oznaczenie: *Fraction of Probes* = FP) mają rozkłady zbliżone do cech oryginalnych i — w założeniu — im mniej z nich zostanie wybranych, tym lepiej.

Do procesu uczenia użyto, dla każdego problemu, połączonych zbiorów: treningowego i walidacyjnego. Podane wyniki uzyskano właśnie dla takiego połączonego zbioru. Użytą metodą szacowania dokładności modelu na niewidzianych danych był 10-krotny test 10-krotnej krosswalidacji (ozn. 10×10CV). Oznacza to, że krosswalidacja wykonywana była 10 razy dla różnych podziałów zbioru treningowego, a na końcu wyniki zostały uśrednione. Rezultaty przedstawione zostały w konwencji A/B/C, gdzie A jest średnim błędem zbalansowanym otrzymanym z krosswalidacji, B — średnim odchyleniem standardowym tego błędu, a C — odchyleniem dla wyników krosswalidacji. Innymi słowy mając serię wyników krosswalidacji (błędów i odchyleń), A i B powstają przez uśrednienie odpowiednich wartości, a C jest odchyleniem standardowym serii błędów.

Dla porównania w tabeli 5.2 zestawiono wyniki najlepszych jak dotychczas modeli zgłoszonych na stronie konkursu.

Zbiór Arcene zawiera dane uzyskane metodą spektrometrii masowej, a zadanie polega na odróżnieniu przypadków rakowych od zdrowych. Najlepszy znaleziony model składał się z selektora rankingowego F-score (patrz str. 20.) wybierającego 8970 cech w połączeniu z klasyfikatorem SVM (wersja liniowa). **10×10CV: 9,2/6,3/1,2**%

Dexter opisuje problem klasyfikacji tekstów, a poszczególne cechy są liczbami wystąpień odpowiednich słów w danym tekście (czyli wektorze da-

Zbiór	Wyniki najlepszych modeli			
	BER	AUC	FF	FP
Arcene	0.0720	0.9811	1.00	0.00
Dexter	0.0325	0.9918	22.50	55.38
Dorothea	0.0854	0.9592	100.00	50.00
Gisette	0.0098	0.9992	15.68	49.23
Madelon	0.0622	0.9378	2.40	0.00

Tabela 5.2: Wyniki najlepszych modeli zamieszczonych na stronie konkursu oraz znalezionych w ramach tej pracy. Objaśnienia skrótów: w tekście.

nych). Najlepsze wyniki udało się uzyskać używając jednopoziomowej selekcji SSV (patrz str. 21.) do 8000 cech połączonej z liniowym SVM. **10×10CV: 5,3/2,6/0,5**%. Ten sam model daje jedynie nieznacznie gorsze wyniki, jeśli ograniczyć liczbę cech do 6650.

Dorothea to jedyny zbiór o wartościach binarnych. Zawiera on dane o aktywności cząsteczek. W tym przypadku ze względu na wyjątkowo dużą liczbę cech (100 000) konieczne było przeprowadzenie wstępnej selekcji polegającej na odrzuceniu atrybutów o małej wariancji. Za takie przyjęto te, które odpowiednio mało razy przyjmowały wartość „1". Użytym progiem było 11 jedynek (na 1150 wektorów w zbiorach: treningowym i walidacyjnym), co było równoznaczne z pozostawieniem 34998 cech. Wybrany model to selekcja na podstawie współczynnika korelacji (patrz str. 18.) do 380 cech i klasyfikator SVM (gaussowski kernel, C=64, bias=0,125). **10×10CV: 15,7/1,9/0,8**%. Innym ciekawym algorytmem selekcji cech (dla tego samego klasyfikatora) jest w tym przypadku selekcja sekwencyjna (patrz str. 12.) składająca się ze wstępnej selekcji do 1000 cech metodą współczynnika korelacji, po której następuje ograniczenie wymiarowości danych do 210 cech przy pomocy selektora rankingowego PCA (głównych składowych: 10, kryterium: suma, szczegóły w rozdziale 3.3). **10×10CV: 16,7/2,1/1,0**%

Dane Gisette opisują pisane odręcznie cyfry 4 i 9, które należy rozróżnić. W tym celu użyto selekcji F-score do 1000 atrybutów w połączeniu z gaussowskim SVM (C=16, bias=0,001). **10×10CV: 1,9/0,5/0,1**%

Zbiór Madelon zawiera dane losowe, wygenerowane przez organizatorów konkursu. Najlepszy znaleziony model to selektor ReliefF z κ=10 (patrz str. 21.) wybierający 20 cech i korzystający z nich SVM (gaussowski ker-

nel, C=2, bias=0,5). **10×10CV: 8,5/1,5/0,3**%. Niewiele gorsze rezultaty osiągnąć można stosując selekcję rankingową PCA (głównych składowych: 20, kryterium: maksimum, szczegóły w rozdziale 3.3) do 20 cech i klasyfikator kNN (k=5). **10×10CV: 8,9/1,7/0,2**% Innym godnym uwagi modelem był 5NN (korzystający z wyników selekcji Relief (do 19 cech). **10×10CV: 9,0/1,7/0,2**%

Zbiór	Selektor	Liczba cech	Klasyfikator
Arcene	F-score	8970	liniowy SVM
Dexter	SSV(one-level)	8000	liniowy SVM
Dorothea	wariancja* + CC	380	gaussowski SVM
Gisette	F-score	1000	gaussowski SVM
Madelon	ReliefF(κ=10)	20	gaussowski SVM

Tabela 5.3: Podsumowanie konfiguracji najlepszych modeli. Wyjaśnienie (*) w tekście.

Podsumowanie

Do głównych owoców tej pracy należy zebranie, po 10-miesiącznych obliczeniach na serwerach Katedry Informatyki Stosowanej, sporej ilości wyników opisujących efektywność wybranych metod selekcji cech w połączeniu z popularnymi klasyfikatorami, na dużych zbiorach danych.

Wyniki te mogą stanowić źródło ciekawych obserwacji, takich jak przedstawione w rozdziale 5.2.2 oraz służyć jako dane porównawcze przy testowaniu nowych metod selekcji. W wykorzystaniu zgromadzonych danych pomocą służyć może program do wizualizacji napisany w ramach tej pracy.

Rozważania na temat komitetów selekcji cech z rozdziału 1.3.1 prowadzą do unifikacji tych metod. Jest ona ciekawa z punktu widzenia teorii, ale również jako wskazówka pomagająca w ich efektywnej implementacji.

Analiza zachowania naiwnego klasyfikatora Bayesowskiego w rozdziale 4 wskazuje na zagrożenie spadku skuteczności tego modelu wraz ze wzrastającą wymiarowością danych. Każda jego implementacja powinna uwzględnić ten fakt i przyjąć jakąś strategię przeciwdziałania utracie dokładności w toku obliczeń, albo przynajmniej sygnalizować, że takie zjawisko ma miejsce.

Zaproponowana w rozdziale 3.3 metoda selekcji rankingowej oparta na analizie czynników głównych (PCA) została sprawdzona na dużych, nietrywialnych zbiorach danych i w niektórych przypadkach osiągnęła wyniki porównywalne z szeroko stosowanymi algorytmami.

Zagadnienie selekcji cech jest problemem szerokim i niektóre jej aspekty zostały jedynie ogólnie omówione bądź zasygnalizowane. Główny nacisk położony został na zadania wysokowymiarowe i stąd taki a nie inny wybór tematyki.

Bibliografia

[1] Włodzisław Duch. Filter methods. Isabelle Guyon, Steve Gunn, Masoud Nikravesh, Lofti Zadeh, redaktorzy, *Feature extraction, foundations and Applications*. Springer, 2005.

[2] Kari Torkkola. Information-theoretic methods for feature selection and construction. Isabelle Guyon, Steve Gunn, Masoud Nikravesh, Lofti Zadeh, redaktorzy, *Feature extraction, foundations and Applications*. Springer, 2005.

[3] Juha Reunanen. Search strategies for wrapper methods. Isabelle Guyon, Steve Gunn, Masoud Nikravesh, Lofti Zadeh, redaktorzy, *Feature extraction, foundations and Applications*. Springer, 2005.

[4] Thomas Navin Lal, Oliver Chapelle, André Elisseeff. Embedded methods. Isabelle Guyon, Steve Gunn, Masoud Nikravesh, Lofti Zadeh, redaktorzy, *Feature extraction, foundations and Applications*. Springer, 2005.

[5] Eugene Tuv. Ensemble learning and feature selection. Isabelle Guyon, Steve Gunn, Masoud Nikravesh, Lofti Zadeh, redaktorzy, *Feature extraction, foundations and Applications*. Springer, 2005.

[6] Krzysztof Grąbczewski, Norbert Jankowski. Mining for complex models comprising feature selection and classification. Isabelle Guyon, Steve Gunn, Masoud Nikravesh, Lofti Zadeh, redaktorzy, *Feature extraction, foundations and Applications*. Springer, 2005.

[7] Andrew Webb. *Statistical Pattern Recognition*. John Wiley and Sons, wydanie 2., 2002.

[8] IEEE Standards Committee 754. *IEEE Standard for binary floating-point arithmetic, ANSI/IEEE Standard 754-1985*. Institute of Electrical and

Electronics Engineers, New York, 1985. Reprinted in ACM SIGPLAN Notices, 22(2):9-25, 1987.