Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych

PRACA DYPLOMOWA MAGISTERSKA
NA KIERUNKU MATEMATYKA

STATYSTYCZNE METODY REGRESJI PORZĄDKOWEJ

Autor:
Marta Sommer

Promotor:
prof. nzw. dr hab.
Przemysław Grzegorzewski

Warszawa, czerwiec 2015

.....

podpis promotora

.....

podpis autora

Spis treści

Wstęp

Regresja porządkowa (ang. *ordinal regression*) jest jednym z działów uczenia maszynowego. Od problemu klasycznej regresji różni ją to, że zmienna odpowiedzi jest dyskretna, natomiast od problemu klasyfikacji to, że zmienna odpowiedzi ma pewien naturalny porządek. Regresja porządkowa zajmuje się zatem uczeniem i oceną jakości predyktora, który modeluje zmienną uporządkowaną i skończoną. Problem regresji porządkowej rozwija się dość szybko m.in. dlatego, że ma on bardzo wiele zastosowań, choćby w systemach rekomendacji, czy bardzo popularnych wyszukiwarkach internetowych. Prześledźmy to na konkretnym przykładzie. Wyobraźmy sobie sytuację, że chcielibyśmy określić, w jakim stopniu danemu człowiekowi spodoba się sprzedawany przez nas produkt. Mamy do dyspozycji zbiór treningowy składający się z wektora zmiennej objaśniającej $\mathbf{x} = (x_1, \ldots, x_d)$, gdzie x_i są różnymi cechami określającymi daną osobę (np. płeć, wiek, wykształcenie, ...). Cechy te – podobnie jak w przypadku zwykłej regresji – mogą być zarówno ciągłe, jak i dyskretne. Mamy również dostęp do zmiennej objaśnianej $\mathbf{y} = (y_1, \ldots, y_r)$, będącej wektorem zero-jedynkowym, wskazującym która klasa została przypisana danemu rekordowi. W naszym przykładzie, zmienną odpowiedzi mogłyby być na przykład: *zdecydowanie mi się nie podoba*, *nie podoba mi się*, *nie mam zdania*, *podoba mi się*, *zdecydowanie mi się podoba*. Widać wyraźnie, że są one uporządkowane.

Najprostszym podejściem do tego typu problemu byłoby zignorowanie kolejności zmiennej odpowiedzi i potraktowanie go, jak zwykłą klasyfikację. W takim przypadku tracimy jednak pewną informację, która prawdopodobnie mogłaby przyczynić się do poprawy naszego klasyfikatora. Idąc w drugą stronę, można potraktować nasz problem, jak zwykłą regresję, zamieniając zmienną odpowiedzi na pewną zmienną ciągłą i to ją modelując, a następnie z powrotem dyskretyzować. Pojawia się tu jednak problem, jak optymalnie zrobić taką transformację, uwzględniając chociażby fakt, że nasza odpowiedź niekoniecznie jest monotoniczna (tzn. np. różnica między *nie podoba mi się* a *nie mam zdania* wcale mnie musi być taka sama, jak między *podoba mi się* a *zdecydowanie mi się podoba*).

Możemy wyróżnić dwa główne nurty w regresji porządkowej:

- prognoza konkretnej obserwacji (nacisk kładziony jest tu na wyznaczenie konkretnego $\mathbf{y}$ dla konkretnego $\mathbf{x}$ np. czy potencjalnemu klientowi spodoba się dany produkt),
- uszeregowanie kilku obserwacji (celem nie jest poznanie estymacji konkretnej zmiennej odpowiedzi, ale takie uszeregowanie kilku rekordów, by te najbardziej preferowane znalazły się na samej górze, a te najmniej na samym dole np. w jakiej kolejności powinny wyświetlić się znalezione strony w wyszukiwarce).

W mojej pracy zajmować się będę przede wszystkim pierwszym punktem, lecz nakreślę też kilka podejść dotyczących drugiego.

Rozdział 1

Opis teoretyczny dostępnych metod

1.1. Postawienie problemu i podstawowe oznaczenia

Na wejściu dany mamy zbiór $\mathcal{D} = (\mathbf{x}^{(i)}, y^{(i)})_{i=1}^{n}$, składający się z n par $(\mathbf{x}, y)$, gdzie:

- $\mathbf{x}^{(i)}$ jest K–wymiarowym wektorem cech (częstym założeniem będzie, że $\mathbf{x}^{(i)} \in \mathbb{R}^K$),
- $y^{(i)}$ jest liczbą symbolizującą kategorię, do której przyporządkowana została i–ta obserwacja, tzn. $y^{(i)} \in \mathcal{Y}$, gdzie $\mathcal{Y} = \{1, \ldots, r\}$ jest zbiorem uporządkowanym według pewnego porządku „$\prec$".

Naszym celem będzie stworzenie modelu, który pozwoli na wybranie najlepszej (nieznanej) kategorii $y_* \in \mathcal{Y}$ dla nowej obserwacji o zadanym wektorze cech $\mathbf{x}_*$.

W tym rozdziale opracujemy kilka rozwiązań, które pozwolą nam się z tym problemem uporać.

1.2. Model proporcjonalnych szans

Najbardziej rozpowszechnionym sposobem modelowania regresji porządkowej jest model proporcjonalnych szans (ang. *proportional odds model*), patrz [1]. Jest to jedna z metod uogólnionych modeli liniowych, bardzo silnie opierająca się na regresji logistycznej. Interesują nas prawdopodobieństwa:

$$\Pi_j(\mathbf{x}) := \mathbb{P}(y = j \mid \mathbf{x}), \quad \text{dla} \quad j = 1, \ldots, r.$$

Idea tej metody polega nie na bezpośrednim szukaniu prawdopodobieństw $\Pi_j(\mathbf{x})$, lecz na wcześniejszym modelowaniu tzw. prawdopodobieństw skumulowanych:

$$\mathbb{P}(y \leq j \mid \mathbf{x}) = \Pi_1(\mathbf{x}) + \ldots + \Pi_j(\mathbf{x}), \quad \text{dla} \quad j = 1, \ldots, r-1.$$

Następnie rozważa się poniższy model logitowy:

$$\log \frac{\mathbb{P}(y \leq j \mid \mathbf{x})}{1 - \mathbb{P}(Y \leq j \mid \mathbf{x})} = \alpha_j + \beta^T \mathbf{x}, \quad \text{dla} \quad j = 1, \ldots, r-1,$$

gdzie $\alpha_j \in \mathbb{R}$ i $\beta \in \mathbb{R}^K$ są parametrami modelu. Należy zauważyć, że parametr β jest stały dla każdego $j = 1, \ldots, r-1$.

Współczynniki modelu – jak w przypadku regresji logistycznej – wyliczamy metodą Raphsona-Newtona, a skumulowane prawdopodobieństwa – po prostym przeliczeniu – dostaniemy ze wzoru:

$$\mathbb{P}(y \leq j \mid \mathbf{x}) = \frac{e^{\alpha_j + \beta^T \mathbf{x}}}{1 + e^{\alpha_j + \beta^T \mathbf{x}}}, \quad \text{dla} \quad j = 1, \ldots, r-1.$$

Szukane prawdopodobieństwa $\Pi_j(\mathbf{x})$ otrzymamy w poniższy sposób:

$$\begin{aligned} \Pi_1(\mathbf{x}) &= \mathbb{P}(Y \leq 1 \mid \mathbf{x}), \\ &\vdots \\ \Pi_i(\mathbf{x}) &= \mathbb{P}(Y \leq i \mid \mathbf{x}) - \mathbb{P}(Y \leq i-1 \mid \mathbf{x}), \\ &\vdots \\ \Pi_r(\mathbf{x}) &= 1 - \mathbb{P}(Y \leq r-1 \mid \mathbf{x}). \end{aligned}$$

Dla nowej obserwacji $\mathbf{x}_*$ wybieramy, oczywiście, tę klasę y_*, która maksymalizuje prawdopodobieństwa $\Pi_j(\mathbf{x}_*)$.

1.3. Wektory maszyn podpierających (SVM)

Wektory maszyn podpierających (ang. *Support Vector Machine*) to bardzo znana i powszechnie stosowana metoda klasyfikacji (patrz [4]). W dużym uproszczeniu, polega ona na konstrukcji dwóch równoległych i maksymalnie oddalonych od siebie hiperpłaszczyzn rozdzielających klasy. By móc obsługiwać przypadki, w których brak liniowej separowalności, wprowadza się dodatkowo karę za nieidealne rozdzielenie klas. W przypadku dwuklasowym (patrz Rys.1.1) budowa modelu sprowadza się do rozwiązania następującego problemu optymalizacyjnego:

$$\min_{\mathbf{w},b} \left\{ \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i \right\},$$

przy ograniczeniach:

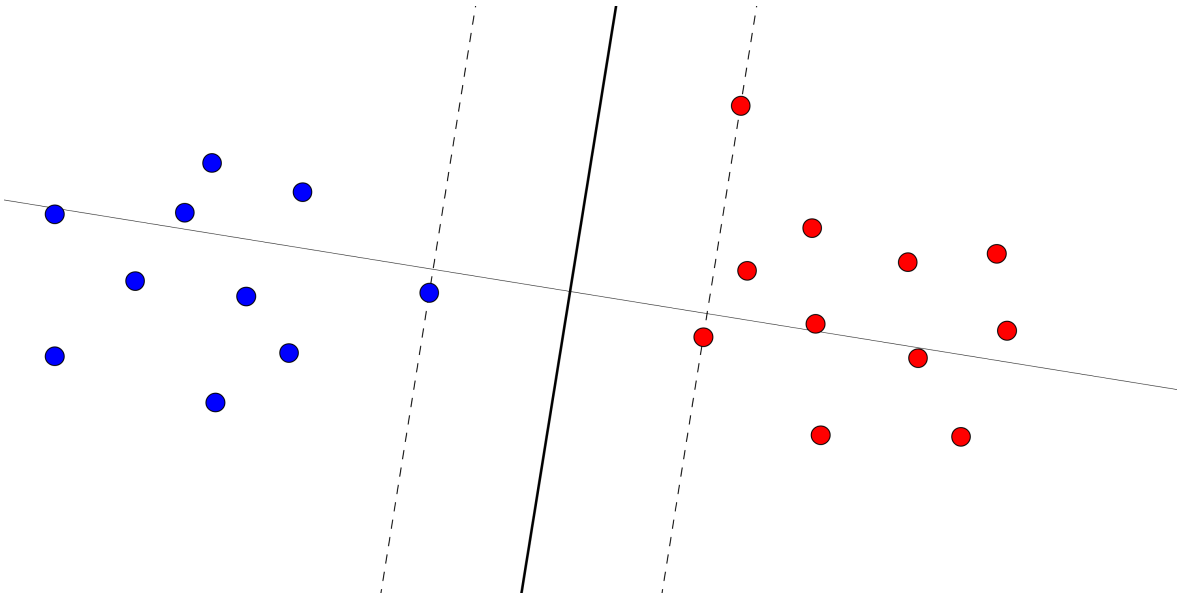
$$\begin{cases} \mathbf{x}_{1i}^T \mathbf{w} + b & \geq +1 - \xi_i, \quad \text{dla} \quad i = 1 \ldots n_1 \\ \mathbf{x}_{2i}^T \mathbf{w} + b & \leq -1 + \xi_i, \quad \text{dla} \quad i = 1 \ldots n_2 \end{cases}$$

gdzie $\mathbf{w} \in \mathbb{R}^K$, $b \in \mathbb{R}$ i $C \in \mathbb{R}$ są parametrami modelu, $\xi_i \geq 0$ dla $i = 1 \ldots n$ są karą mierzoną dla każdej obserwacji przy ustalonej hiperpłaszczyźnie, $\mathbf{x}_{1i}$ oznacza wektor cech obserwacji należących do klasy pierwszej, a $\mathbf{x}_{2i}$ wektor cech obserwacji należących do klasy drugiej, zaś n_1 i n_2 to liczności tych klas.

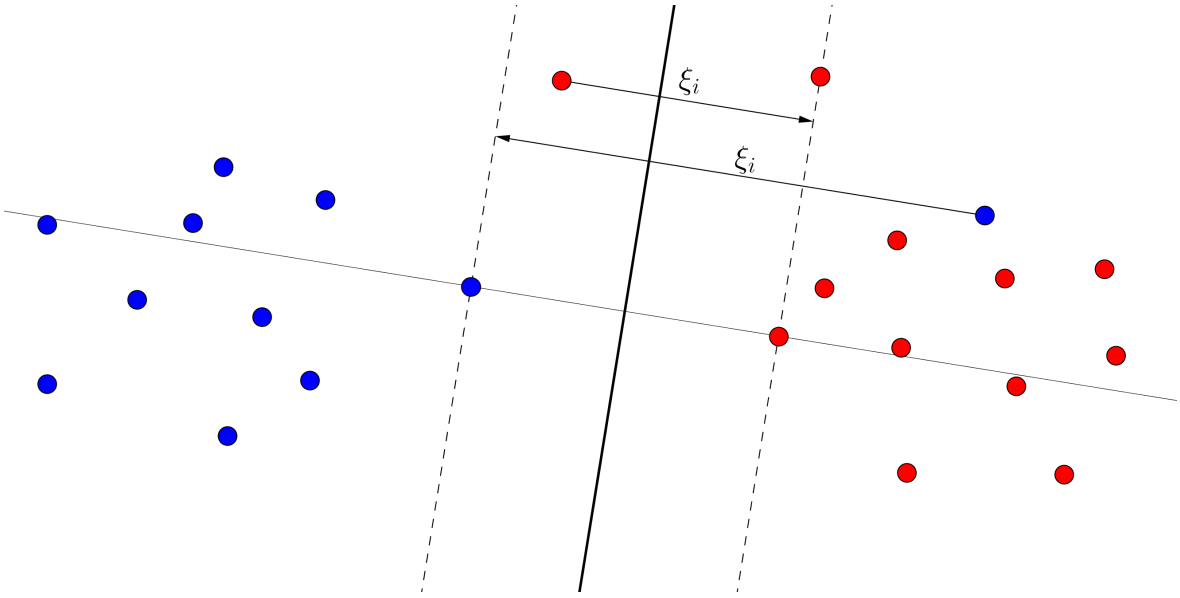
Tak to wygląda w przypadku klasyfikacji dwuklasowej. Przyjrzyjmy się teraz, jak w łatwy sposób można zaadaptować powyższą metodę do rozważanej przez nas regresji porządkowej.

Tym razem, powołując się na [1] i [5], będziemy rozwiązywać następujący problem optymalizacyjny:

$$\min_{\mathbf{w},b_1,\ldots,b_{r-1}} \left\{ \frac{1}{2}||\mathbf{w}||^2 + C \sum_{j=1}^{r-1} \left(\sum_{k=1}^{j} \sum_{i=1}^{n_k} \xi_{ki}^{j} + \sum_{k=j+1}^{r} \sum_{i=1}^{n_k} \xi_{ki}^{*j} \right) \right\},$$

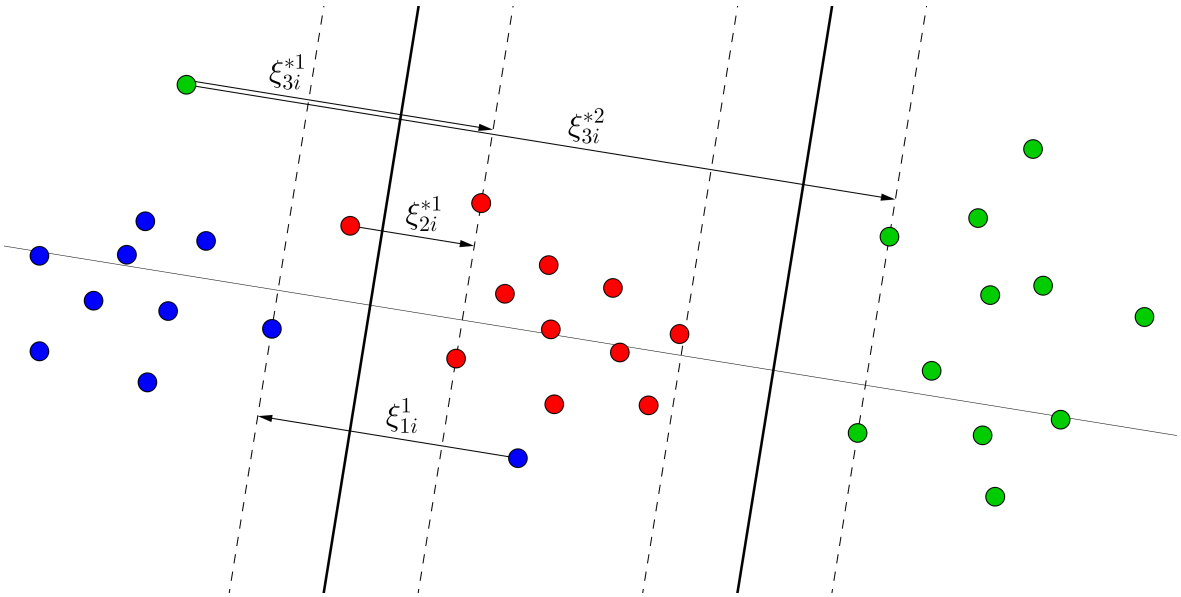


(a) Przypadek separowalny.



(b) Przypadek nieseparowalny.

Rysunek 1.1: Przykładowe rozdzielenie klas metodą SVM. W przypadku nieseparowalnym dokładamy karę, będącą odległością źle zaklasyfikowanej obserwacji od odpowiedniego marginesu.



Rysunek 1.2: Przykładowa klasyfikacja metodą SVM.

przy ograniczeniach:

$$\begin{cases} \mathbf{x}_{ki}^T\mathbf{w} - b_j & \leq -1 + \xi_{ki}^j, \qquad \text{dla} \quad k = 1, \ldots, j \quad \text{oraz} \quad i = 1, \ldots, n_k \\ \mathbf{x}_{ki\mathbf{w}}^T - b_j & \geq +1 - \xi_{ki}^{*j}, \qquad \text{dla} \quad k = j+1, \ldots, r \quad \text{oraz} \quad i = 1, \ldots, n_k, \end{cases}$$

gdzie $\mathbf{w} \in \mathbb{R}^K, b_1 \in \mathbb{R}, \ldots, b_{r-1} \in \mathbb{R}, C \in \mathbb{R}$ są parametrami modelu, $\mathbf{x}_{ki}^T$ oznacza $i-$tą obserwację należącą do $k-$tej klasy, n_k to liczność $k-$tej klasy, $j = 1 \ldots r-1$, a ξ to kary, których konstrukcję wyjaśnimy poniżej.

Przyjrzyjmy się, czym różni się nasz nowy problem od problemu optymalizacyjnego w standardowej klasyfikacji. Przede wszystkim – podobnie jak w modelu proporcjonalnych szans – mamy tu do czynienia z $(r-1)$–hiperpłaszczyznami, rzutowanymi na jeden, wspólny dla wszystkich obserwacji, kierunek $\mathbf{w}$. Przy wyznaczaniu kolejnych hiperpłaszczyzn, bierzemy pod uwagę wszystkie klasy. Kary naliczane są więc w następujący sposób (patrz rysunek 1.2). Dla ustalonego progu b_j obserwujemy wartości funkcji $x_{ki}^T\mathbf{w}$. Dla obserwacji z niższych klas (tzn. klas $1, \ldots, j$), wartości te powinny być niższe niż dolna granica $b_j - 1$. Jeśli tak nie jest, wtedy jako błąd próbki ξ_{ki}^j dla progu b_j uznaje się ξ_{ki}^j, czyli odległość tego punktu od rozpatrywanej dolnej granicy. Analogicznie, dla obserwacji z wyższych klas wartości $\mathbf{w}x_{ki}$ powinny być wyższe niż górna granica $b_j + 1$. Jeśli tak nie jest, to otrzymujemy błędy ξ_{ki}^{*j}.

Budowa modelu i tym razem sprowadza się więc do problemu optymalizacyjnego. Wyznaczywszy, przy użyciu pewnego algorytmu, szukane parametry, dostaniemy równania $r-1$ hiperpłaszczyzn:

$$\begin{cases} \mathbf{x}^T\mathbf{w} - b_1 & = 0 \\ & \vdots \\ \mathbf{x}^T\mathbf{w} - b_{r-1} & = 0 \end{cases}$$

Dla nowej obserwacji $\mathbf{x}_*$ wystarczy policzyć $\mathbf{x}_*^T\mathbf{w}$ i sprawdzić między którymi dwoma hiperpłaszczyznami się znajduje i przypisać jej odpowiednią klasę.

1.4. Sieci neuronowe

Sieci neuronowe to bardzo proste i szeroko stosowane narzędzie zarówno w problemach regresji, jak i klasyfikacji. Znalazło ono również swoje zastosowanie w regresji porządkowej (por. [3]).

Standardowo, na wejściu otrzymujemy zbiór uczący w postaci n par $(\mathbf{x}, y)$, gdzie $\mathbf{x} = (x_1, \ldots, x_K)^T$ jest wektorem cech, a y numerem klasy. Tym razem jednak, dodatkowo modyfikujemy zmienną odpowiedzi w taki sposób, by zamiast liczby rzeczywistej otrzymać zero-jedynkowy wektor odpowiedzi $\mathbf{y} = (y_1, \ldots, y_r)^T$ reprezentujący klasę, do której należy dana obserwacja, tzn. $y_i = \mathbb{I}\{y = i\}$.

W przeciwieństwie do zwykłej klasyfikacji, nasza sieć neuronowa będzie zakładać porządek zmiennej odpowiedzi. W jaki sposób? Mianowicie, jako wektor wyjściowy, zamiast wektora $\mathbf{y} = (\underbrace{0, \ldots, 0}_{i-1}, 1, \ldots, 0)^T$, mającego jedynkę na i–tym miejscu, jeśli obserwacja należała do i–tej klasy, rozważać będziemy wektor $\mathbf{y} = (\underbrace{1, \ldots, 1}_{i}, 0, \ldots, 0)^T$, mający jedynki na miejscach od pierwszego do i–tego.

Otrzymujemy w ten sposób sieć neuronową o K neuronach w warstwie wejściowej (z których każdy reprezentuje inną cechę z wektora $\mathbf{x}$), jednej (bądź więcej) warstwie ukrytej o m neuronach i warstwie wyjściowej, zawierającej r neuronów, które reprezentują odpowiedź $\mathbf{y}$ w formie opisanej powyżej. Za funkcję przejścia przyjmiemy funkcję sigmoidalną $f(x) = \frac{1}{1+e^{-x}}$, dobrze reprezentującą przynależność do danej klasy jako prawdopodobieństwo.

Uczenie sieci neuronowej będzie się odbywało algorytmem propagacji wstecznej z kwadratową funkcją straty (można też użyć jakiejś innej, np. entropii). W dużym uproszczeniu, algorytm wygląda następująco (szczegóły każdego kroku będą wyjaśnione poniżej):

1. Wybieramy małe wagi początkowe oraz pewnie niewielki współczynnik $\eta > 0$.
2. Losujemy parę $(\mathbf{x}, \mathbf{y})$ ze zbioru uczącego.
3. Przebiegamy sieć w przód.
4. Przebiegamy sieć w tył (licząc błąd dla każdego neuronu).
5. Zmieniamy wagi.
6. Dopóki nie osiągniemy zadowalająco niskiego błędu, wracamy do punktu 2).

Wyjaśnijmy teraz ważniejsze punkty powyższego algorytmu.

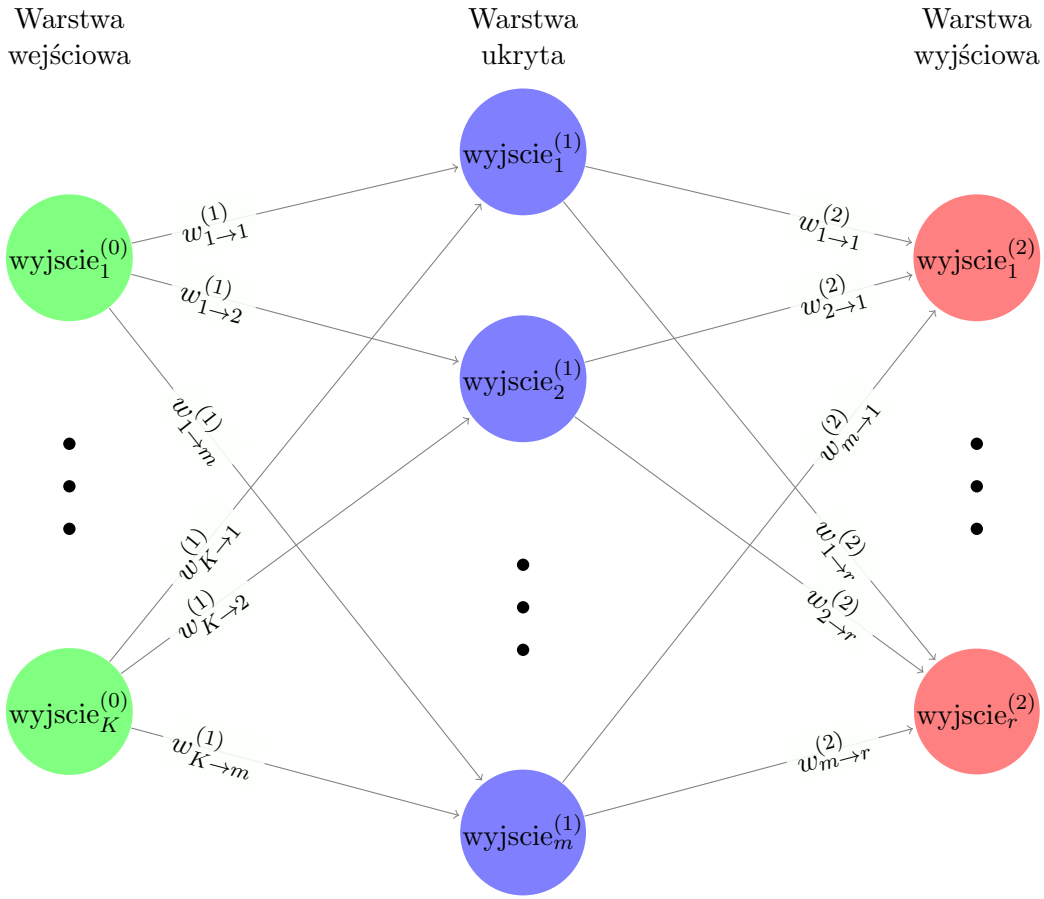
Ad. 3)

Dla każdego neuronu obliczamy wartość wejściową ze wzoru (patrz Rys.1.3):

$$wejscie_j^{(i)} = \sum_{k:\ \exists\ w_{k\to j}^{(i)}} \left(w_{k\to j}^{(i)} \cdot wyjscie_k^{(i-1)} \right),$$

gdzie $wyjscie_i^{(j)}$ to wartość i–tego neuronu w j–tej warstwie, a $wyjscie_i^{(0)} = x_i$, zaś $w_{k\to j}^{(i)}$ oznacza wagę między k–tym neuronem $(i-1)$–warstwy i j–tym neuronem i-tej warstwy. Następnie wyznaczamy wartość wyjściową:

$$wyjscie_j^{(i)} = f\left(wejscie_j^{(i)} \right),$$



Rysunek 1.3: Przykładowa sieć neuronowa.

gdzie $f(\cdot)$ to wybrana przez nas funkcja sigmoidalna – w naszym przypadku $f(x) = \frac{1}{1+e^{-x}}$.

Ad. 4)

Dla warstwy wyjściowej błąd ma postać:

$$\delta_j = wyjscie_j^{(i)} \cdot \left(1 - wyjscie_j^{(i)}\right) \cdot (wyjscie_j^{(i)} - y_j),$$

zaś dla warstw ukrytych:

$$\delta_j^{(i)} = wyjscie_j^{(i)} \cdot \left(1 - wyjscie_j^{(i)}\right) \cdot \sum_{k:\ \exists\ w_{j\to k}^{(i+1)}} \left(w_{j\to k}^{(i+1)} \cdot \delta_k^{(i-1)}\right).$$

Ad. 5)

Modyfikacja wag przebiega następująco:

$$w_{k\to j}^{(i)} := w_{k\to j}^{(i)} - \eta \cdot \delta_j^{(i)} \cdot wyjscie_k^{(i-1)}.$$

Predykcja polega już tylko na przejściu algorytmu w przód z nowymi obserwacjami wejściowymi $\mathbf{x}_*$ i ustaleniu progu (najczęściej równego $0,5$, gdyż wartość neuronu wyjściowego reprezentuje pewne prawdopodobieństwo), klasyfikującego neuron wyjściowy jako jedynkę. Skanujemy

wektor wyjściowy zaczynając od y_1 i kończymy, gdy pierwszy raz natkniemy się na 0. Przypisujemy obserwacji taką klasę, jaką długość miał znaleziony przez nas ciąg jedynek. Może się zdarzyć, że wyjściowy wektor nie będzie ciągiem malejącym, tzn. zamiast łatwo interpretowalnego wektora $(1, \ldots, 1, 0, \ldots, 0)$ otrzymamy na przykład wektor $(1, 1, 0, 1, 1, 1, 0, \ldots, 0)$, co trochę przeczy intuicji, bo sugeruje, że obserwacja należy do klasy czwartej, piątej i szóstej, ale do trzeciej już nie. W takim wypadku, tak jak zostało to opisane powyżej, zaklasyfikowalibyśmy ją do klasy drugiej, przymykając niejako oko na to, co dzieje się później.

1.5. Metoda Franka i Halla

Podejście zaproponowane przez E. Franka i M. Halla (por. [2]) do zagadnienia regresji porządkowej jest nieco inne, niż w metodach przedstawionych do tej pory metody. Polega bowiem nie na stworzeniu nowego modelu, ale na odpowiednim przedefiniowaniu zbioru danych, a następnie na sprowadzeniu zadania do problemu zwykłej klasyfikacji z dwiema klasami. Dokładniej, przekształcamy r–klasowy model regresji porządkowej do $(r-1)$ dwuklasowych problemów klasyfikacji.

Uproszczony algorytm budowy modelu wygląda następująco:

1. Modyfikujemy zbiór uczący (otrzymując $r-1$ nowych zbiorów uczących).
2. Dla każdego nowo uzyskanego zbioru danych dopasowujemy zwykły model klasyfikacyjny (np. drzewo klasyfikacyjne) taki, który zwraca prawdopodobieństwa przynależności do poszczególnych klas.
3. Robimy predykcję dla nowej obserwacji.

Przyjrzyjmy się teraz kolejnym krokom algorytmu dokładniej.

Ad. 1)

Chcemy otrzymać $r-1$ nowych zbiorów o zero-jedynkowej zmiennej odpowiedzi. W jaki sposób to zrobić? Macierz atrybutów pozostaje bez zmian, a zmienia się jedynie wektor zmiennej odpowiedzi (patrz Rys.1.4) według zasady:

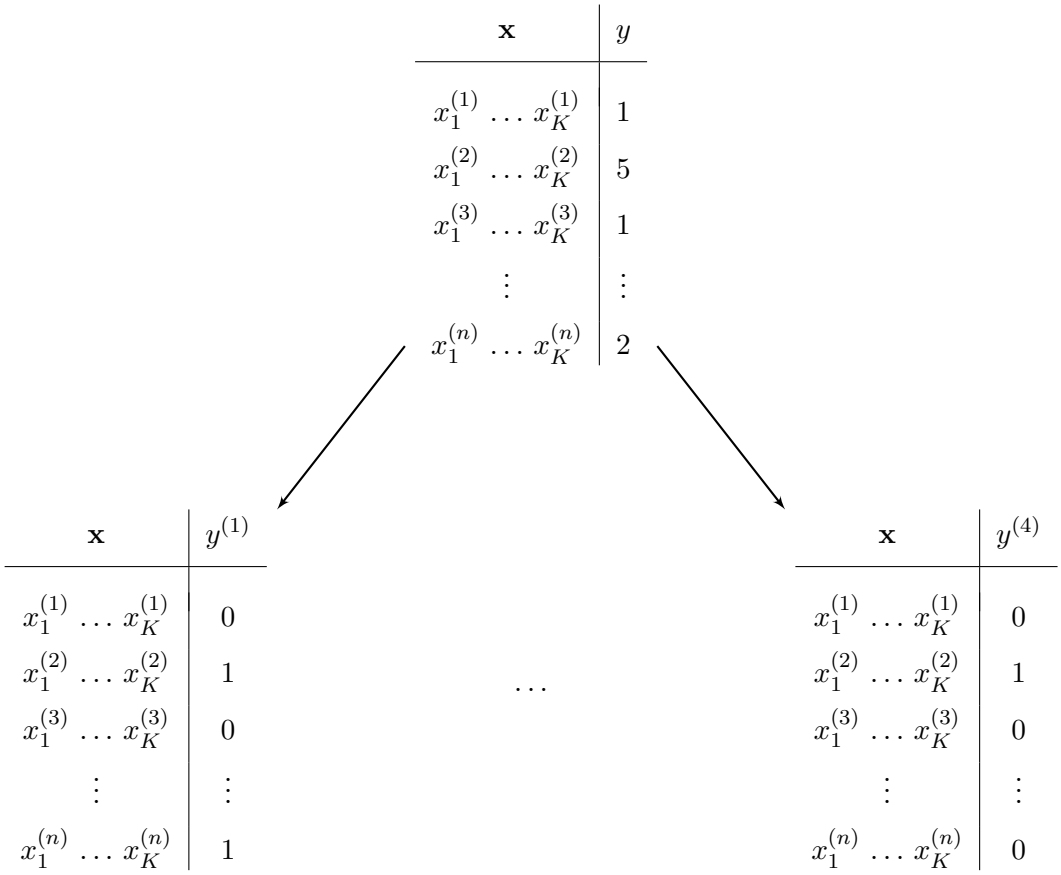
$$\begin{aligned} y_i^{(1)} &= \mathbb{I}\{y_i > 1\} \\ &\vdots \\ y_i^{(r-1)} &= \mathbb{I}\{y_i > r-1\}, \end{aligned}$$

gdzie $y_i^{(j)}$ to i-ta odpowiedź w j-tym, utworzonym przez nas, zbiorze oraz $i = 1, \ldots, n$, zaś $j = 1, \ldots, r-1$.

Ad. 3)

Dla nowego wektora atrybutów $\mathbf{x}$ robimy predykcję na $r-1$ modelach uzyskanych w punkcie drugim. Zwracamy jednak nie predykcję klasy, ale prawdopodobieństwo przynależności do klasy oznaczonej przez nas jako pierwszej. Uzyskujemy w ten sposób $r-1$ następujących prawdopodobieństw: $\mathbb{P}(y > 1), \ldots, \mathbb{P}(y > r-1)$.

Nas natomiast interesują prawdopodobieństwa: $\mathbb{P}(y = 1), \ldots, \mathbb{P}(y = r-1)$,



Rysunek 1.4: Modyfikacja przykładowego zbioru uczącego.

które łatwo otrzymamy, korzystając z następującego wzoru łańcuchowego:

$$\begin{aligned}
\mathbb{P}(y=1) &= 1-\mathbb{P}(y>1)\\
&\vdots\\
\mathbb{P}(y=i) &= \mathbb{P}(y>i-1)-\mathbb{P}(y>i) \qquad \text{dla} \quad i=2,\dots,r-1\\
&\vdots\\
\mathbb{P}(y=r) &= \mathbb{P}(y>r-1).
\end{aligned}$$

Ostatecznie, nowej obserwacji przypisujemy tę klasę, której prawdopodobieństwo $\mathbb{P}(y=i)$ było największe.

1.6. Procesy gaussowskie

Kolejną metodą modelowania problemu regresji porządkowej jest użycie procesu gaussowskiego. Jest to metoda popularna szczególne przy zwykłej regresji, znalazła ona jednak również zastosowanie w klasyfikacji zarówno jedno, jak i wieloetykietowej. Chu i Ghahramani w pracy [6] pokazują, jak rozszerzyć ją na regresję porządkową.

Pomysł modelowania regresji porządkowej polega na wprowadzeniu tzw. zmiennej ukrytej, będącej niejako krokiem pośrednim w modelowaniu zmiennej odpowiedzi. Mianowicie, zamiast dawać od razu odpowiedź, do której klasy przypisujemy daną obserwację, próbujemy ją najpierw scharakteryzować jako liczbę rzeczywistą, by móc ją niejako umieścić na prostej. Dzięki temu uzyskamy pewne uporządkowanie między naszymi obserwacjami, by następnie wybrać progi, które będą już klasyfikować obserwacje. W celu wyrobienia sobie intuicji przeanalizujmy całe rozumowanie „od tyłu", zaczynając od predykcji. Podstawowym założeniem *a priori* tej metody jest to, że zmienna ukryta F jest procesem gaussowskim tzn., że jej rozkłady skończenie wymiarowe są normalne. Pełną charakteryzację takiego procesu tworzą dwie informacje – średnia (standardowo przyjmuje się 0) oraz macierz kowariancji $\mathbf{\Sigma}$. Dla celów tej pracy przyjmiemy, że elementy macierzy kowariancji definiowane są w następujący sposób:

$$\mathbf{\Sigma}_{ij} = \mathbf{\Sigma}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left\{-\frac{\kappa}{2}\sum_{\xi=1}^{K}(x_\xi^{(i)} - x_\xi^{(j)})^2\right\},$$

gdzie $\kappa > 0$, a $x_\xi^{(i)}$ oznacza ξ-ty element wektora $\mathbf{x}^{(i)}$. Zatem $F|\mathbf{X} \sim \mathcal{N}(0, \mathbf{\Sigma})$, czyli:

$$\mathbb{P}(\mathbf{f}|\mathbf{X}) = \frac{1}{(2\Pi)^{\frac{n}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\mathbf{f}^T\mathbf{\Sigma}^{-1}\mathbf{f}\right\}, \tag{1.1}$$

gdzie $\mathbf{f} = [f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(n)})]^T$ to wektor zawierający realizację zmiennej ukrytej, odpowiadający kolejnym obserwacjom ze zbioru uczącego.

Wyobraźmy sobie teraz, że dopasowaliśmy model i znamy wszystkie niezbędne parametry. W uproszczony sposób predykcja wygląda następująco:

1. na wejściu otrzymujemy nową obserwację o danym wektorze cech $\mathbf{x}_*$,
2. w pewien sposób wyliczamy dla niej liczbę rzeczywistą $f(\mathbf{x}_*)$,
3. za pomocą przekształcenia prostej rzeczywistej na r podzbiorów, wyznaczamy najlepszy y_*.

A teraz prześledźmy wszystko krok po kroku. Interesuje nas wyznaczenie y_*, dla którego prawdopodobieństwo $\mathbb{P}(y_*|\mathbf{X}, \mathbf{y}, x_*)$ jest największe. Za pomocą zmiennej ukrytej rozpiszmy je w następujący sposób:

$$\mathbb{P}(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \int_{\mathbb{R}} \mathbb{P}(y_*|f(\mathbf{x}_*))\mathbb{P}(f(\mathbf{x}_*)|\mathbf{X}, \mathbf{y})df(\mathbf{x}_*). \tag{1.2}$$

Analogicznie:

$$\mathbb{P}(f(\mathbf{x}_*)|\mathbf{X}, \mathbf{y}) = \int_{\mathbb{R}^n} \mathbb{P}(f(\mathbf{x}_*)|\mathbf{f})\mathbb{P}(\mathbf{f}|\mathbf{X}, \mathbf{y})d\mathbf{f}. \tag{1.3}$$

By móc policzyć powyższe całki, poszukamy kolejno prawdopodobieństw, których iloczyny je tworzą.

Korzystając z informacji, że zmienna f jest procesem gaussowskim, czyli:

$$\begin{bmatrix} \mathbf{f} \\ f(\mathbf{x}_*) \end{bmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma} & \mathbf{\Sigma}_* \\ \mathbf{\Sigma}_*^T & \Sigma_{**} \end{pmatrix}\right],$$

gdzie $\mathbf{\Sigma}_* = [\Sigma(\mathbf{x}_1, \mathbf{x}_*), \ldots, \Sigma(\mathbf{x}_n, \mathbf{x}_*)]^T$, a $\Sigma_{**} = \Sigma(x_*, x_*)$, otrzymujemy, że:

$$f(x_*)|\mathbf{f} \sim \mathcal{N}(\mathbf{f}^T\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_*, \Sigma_{**} - \mathbf{\Sigma}\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_*). \tag{1.4}$$

Zajmijmy się teraz prawdopodobieństwem $\mathbb{P}(\mathbf{f}|\mathbf{X},\mathbf{y})$. Korzystając z podejścia bayesowskiego (patrz A.1), rozpiszmy je jako:

$$\mathbb{P}(\mathbf{f}|\mathbf{X},\mathbf{y}) = \frac{\mathbb{P}(\mathbf{y}|\mathbf{f})\mathbb{P}(\mathbf{f}|\mathbf{X})}{\mathbb{P}(\mathbf{y}|\mathbf{X})}. \tag{1.5}$$

Znamy już $\mathbb{P}(\mathbf{f}|\mathbf{X})$ – jest to prawdopodobieństwo *a priori* (1.1). $\mathbb{P}(\mathbf{y}|\mathbf{X})$, jako stała niezależna od $\mathbf{f}$, nie jest nam potrzebne do wyznaczenia $\hat{\mathbf{f}}$. Zostawmy je więc na razie i wróćmy do niego później, kiedy będziemy estymować parametry modelu. Zostaje nam więc do znalezienia $\mathbb{P}(\mathbf{y}|\mathbf{f})$, tzw. wiarogodność. Ponieważ wszystkie obserwacje są niezależne, otrzymujemy:

$$\mathbb{P}(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} \mathbb{P}(y^{(i)}|f(\mathbf{x}^{(i)})). \tag{1.6}$$

Gdybyśmy zakładali idealną sytuację, wtedy $\mathbb{P}_{ideal}(y^{(i)}|f(\mathbf{x}^{(i)})) = \mathbb{I}_{\{f(\mathbf{x}^{(i)})\in(b_{y^{(i)}-1},b_{y^{(i)}}]\}}$, gdzie $b_0 = -\infty, b_r = +\infty$, a $b_i \in \mathbb{R}$ dla $i = 1,\ldots,r-1$ to parametry modelu. Wygodniej, można b_i sparametryzować jako: $b_1 \in \mathbb{R}$ oraz $b_i = \sum_{t=2}^{j} \Delta_t + b_1$, gdzie $\Delta_t > 0$ oraz $j = 2,\ldots,r-1$. Bardzo rzadko mamy jednak do czynienia z sytuacją idealną, dlatego będziemy budować model, zakładając dodatkowy szum δ o rozkładzie $\mathcal{N}(0,\sigma^2)$. Wtedy prawdopodobieństwo zmienia się następująco:

$$\begin{aligned}
\mathbb{P}(y^{(i)}|f(\mathbf{x}^{(i)})) &= \int_{\mathbb{R}} \mathbb{P}_{ideal}(y^{(i)},\delta_i|f(\mathbf{x}^{(i)}))d\delta_i = \int_{\mathbb{R}} \mathbb{P}_{ideal}(y^{(i)}|f(\mathbf{x}^{(i)}),\delta_i)\mathbb{P}(\delta_i)d\delta_i = \\
&= \int_{\mathbb{R}} \mathbb{P}(\delta_i)\mathbb{I}_{\{f(\mathbf{x}^{(i)})+\delta_i\in(b_{y^{(i)}-1},b_{y^{(i)}}]\}}d\delta_i = \\
&= \Phi\left(\frac{b_{y^{(i)}} - f(x^{(i)})}{\sigma}\right) - \Phi\left(\frac{b_{y^{(i)}-1} - f(x^{(i)})}{\sigma}\right),
\end{aligned} \tag{1.7}$$

gdzie $\Phi(\cdot)$ to dystrybuanta standardowego rozkładu normalnego.

Przejdźmy teraz do szukania najlepszej estymacji $\hat{\mathbf{f}}$. Zdefiniujmy $S(\mathbf{f}) := -\ln \mathbb{P}(\mathbf{f}|\mathbf{X},\mathbf{y})$, wtedy:

$$\hat{\mathbf{f}} := \underset{\mathbf{f}}{\operatorname{argmax}}\{\mathbb{P}(\mathbf{f}|\mathbf{X},\mathbf{y})\} = \underset{\mathbf{f}}{\operatorname{argmax}}\{\ln \mathbb{P}(\mathbf{f}|\mathbf{X},\mathbf{y})\} = \underset{\mathbf{f}}{\operatorname{argmin}}\{-\ln \mathbb{P}(\mathbf{f}|\mathbf{X},\mathbf{y})\} = \underset{\mathbf{f}}{\operatorname{argmin}}\{S(\mathbf{f})\}.$$

Korzystając z równań (1.1), (1.5) i (1.6) można łatwo zobaczyć, że:

$$S(\mathbf{f}) \propto \sum_{i=1}^{n} l(y^{(i)}, f(\mathbf{x}^{(i)})) + \frac{1}{2}\mathbf{f}^T\boldsymbol{\Sigma}^{-1}\mathbf{f},$$

gdzie $l(y^{(i)}, f(\mathbf{x}^{(i)})) := -\ln \mathbb{P}(y^{(i)}|f(\mathbf{x}^{(i)}))$. Nie da się znaleźć minimum tej funkcji analitycznie. Natomiast, żeby uzyskać najlepsze przybliżenie $\hat{\mathbf{f}}$, wystarczy zastosować do funkcji $S(\mathbf{f})$ dowolny algorytm optymalizacyjny (np. algorytm Newtona-Raphsona).

Przypomnijmy, że naszym celem jest w tej chwili wyznaczenie $\mathbb{P}(\mathbf{f}|\mathbf{X},\mathbf{y})$. Ponieważ później będziemy chcieli obliczyć całkę (1.3), nie wystarczy nam jedynie estymator $\hat{\mathbf{f}}$, wygodnie byłoby dla nas, gdyby to prawdopodobieństwo okazało się gaussowskie. Da się to osiągnąć dzięki przybliżeniu Laplace'a.

Na początku zauważmy, że:

$$\frac{\partial^2 S(\mathbf{f})}{\partial \mathbf{f}\partial \mathbf{f}^T} = \boldsymbol{\Sigma}^{-1} + \boldsymbol{\Lambda},$$

gdzie

$$\mathbf{\Lambda} = \begin{bmatrix} \frac{\partial^2 l(y^{(1)}, f(\mathbf{x}^{(1)}))}{\partial^2 f(\mathbf{x}^{(1)})} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial^2 l(y^{(n)}, f(\mathbf{x}^{(n)}))}{\partial^2 f(\mathbf{x}^{(n)})} \end{bmatrix}.$$

Rozwijając funkcję $S(\mathbf{f})$ w szereg Taylora w punkcie $\hat{\mathbf{f}}$ i pamiętając, że $S'(\hat{\mathbf{f}}) = 0$, otrzymamy następujące przybliżenie:

$$S(\mathbf{f}) = S(\hat{\mathbf{f}}) + \frac{1}{2}(\mathbf{f} - \hat{\mathbf{f}})^T(\mathbf{\Sigma}^{-1} + \hat{\mathbf{\Lambda}})(\mathbf{f} - \hat{\mathbf{f}}),$$

gdzie $\hat{\mathbf{\Lambda}}$ jest macierzą $\mathbf{\Lambda}$ wyznaczoną dla $\hat{\mathbf{f}}$. Z powyższego równania bezpośrednio wynika, że:

$$F|\mathbf{X}, \mathbf{y} \sim \mathcal{N}\left(\hat{\mathbf{f}}, (\mathbf{\Sigma}^{-1} + \hat{\mathbf{\Lambda}})^{-1}\right). \tag{1.8}$$

Tak więc zostało nam już tylko zoptymalizowanie $\mathbb{P}(\mathbf{y}|\mathbf{X})$ tak, by wyznaczyć najlepszy wektor parametrów $\mathbf{\Theta} = [\kappa, \sigma, b_1, \Delta_2, \ldots, \Delta_{r-1}]^T$, który przyda nam się przy predykcji. Znów, odwołując się do przybliżenia Laplace'a i do faktu, że

$$\mathbb{P}(\mathbf{y}|\mathbf{X}) = \int \mathbb{P}(\mathbf{y}|\mathbf{f}, \mathbf{X})\mathbb{P}(\mathbf{f}|\mathbf{X})d\mathbf{f},$$

otrzymujemy

$$\mathbb{P}(\mathbf{y}|\mathbf{X}) \simeq e^{-S(\hat{\mathbf{f}})}\left|\mathbf{I} + \mathbf{\Sigma}\hat{\mathbf{\Lambda}}\right|^{-\frac{1}{2}},$$

gdzie $\mathbf{I}$ jest macierzą jednostkową $n \times n$. Bez problemu możemy teraz znaleźć maksimum prawdopodobieństwa $\mathbb{P}(\mathbf{y}|\mathbf{X})$ iteracyjnie lub nawet analitycznie (por.[6]).

Wróćmy teraz do szukanych całek (1.2) i (1.3). Korzystając z równań (1.4) i (1.8) dostaniemy, że w przybliżeniu:

$$f(\mathbf{x}_*)|\mathbf{X}, \mathbf{y} \sim \mathcal{N}(\mu_*, \sigma_*^2),$$

gdzie $\mu_* = \mathbf{\Sigma}^T\mathbf{\Sigma}^{-\mathbf{1}}\hat{\mathbf{f}}$ oraz $\sigma_*^2 = \Sigma_{**} - \mathbf{\Sigma}^T(\mathbf{\Sigma} + \hat{\mathbf{\Lambda}}^{-\mathbf{1}})^{-1}\mathbf{\Sigma}$. Natomiast, korzystając jeszcze z równania (1.7) oraz posiłkując się dowodem A.2, otrzymujemy rozkład predykcyjny następującej postaci:

$$\mathbb{P}(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \Phi\left(\frac{b_{y_*} - \mu_*}{\sqrt{\sigma^2 + \sigma_*^2}}\right) - \Phi\left(\frac{b_{y_*-1} - \mu_*}{\sqrt{\sigma^2 + \sigma_*^2}}\right).$$

Dla nowej obserwacji wystarczy teraz jedynie wyznaczyć $\underset{i}{\text{argmax}}\mathbb{P}(y_* = i|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$.

Rozdział 2

Diagnostyka modelu

W poprzednim rozdziale poznaliśmy kilka metod rozwiązania problemu regresji porządkowej. Teraz chcielibyśmy dowiedzieć się, która z nich jest najlepsza. Oczywiście nie da się stwierdzić tego w ogólności, gdyż skuteczność metod zależy od konkretnych danych. Mamy jednak do dyspozycji kilka wskaźników, które pomogą nam w ocenie jakości modelu. Nie różnią się one zbytnio od tych, które znamy ze zwykłej klasyfikacji – są jedynie ich pewnym uogólnieniem. Możemy zatem używać:

- procentu poprawnej klasyfikacji,
- średniego błędu bezwzględnego (lub kwadratowego),
- krzywej ROC i współczynnika AUC.

2.1. Procent poprawnej klasyfikacji

Procent poprawnej klasyfikacji jest bardzo prosty i intuicyjny. Żeby nie było jednak żadnych wątpliwości zdefiniujmy go formalnie. Niech y będzie n-wymiarowym wektorem prawdziwych klas dla zbioru testowego, a y^* wektorem klas otrzymanych z modelu dla tego zbioru. Wówczas procentem poprawnej klasyfikacji nazywamy:

$$PPK = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{y_i = y_i^*\}. \tag{2.1}$$

Niestety, współczynnik ten ma wiele wad.

Po pierwsze, działa źle w przypadku, gdy klasy są niezbilansowane. Zobaczmy to na przykładzie, załóżmy, że rozpatrujemy przypadek trzy-klasowy. Klasa pierwsza występuje w 95% przypadków, a klasa druga i trzecia w 5% przypadków. Załóżmy również, że nasz klasyfikator jest bardzo prymitywny i klasyfikuje wszystko jako jedynki, nie zważając na wektor cech. Co wtedy otrzymujemy? Aż 95% procent poprawnej klasyfikacji!

Po drugie, traktuje każdy błąd zero-jedynkowy. Na przykład, gdy rozważamy regresję porządkową z dziesięcioma klasami, PPK tak samo traktować będzie przypisanie obserwacji jedynki i dziewiątki, gdy rzeczywiście była dziesiątka. A przecież to jest ogromna różnica.

2.2. Średni błąd bezwzględny

Drugą wymienioną wyżej wadę procentu poprawnej klasyfikacji omija tzw. średni błąd bezwzględny. Korzystając z tych samych oznaczeń, możemy go zdefiniować jako:

$$ABSerr = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^*|. \tag{2.2}$$

Wciąż jednak nie jest on odporny na niezbilansowane klasy.

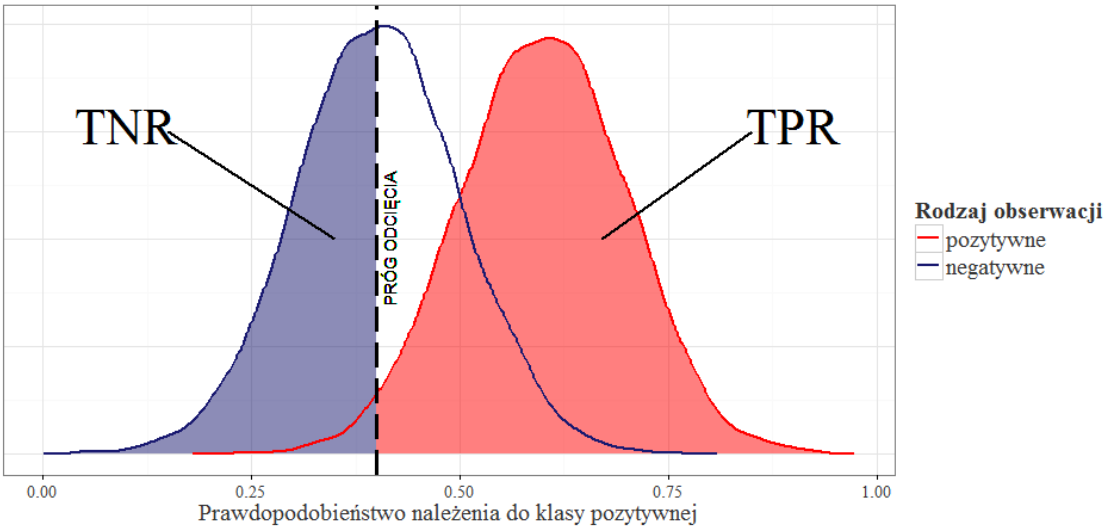
2.3. Krzywa ROC w przypadku dwuklasowym

W przypadku dwuklasowym najczęściej stosowaną metodą oceny modelu jest porównywanie krzywych ROC (ang. *Receiver Operating Characteristic*) i pól pod tą krzywą, czyli AUC (ang. *Area Under the Curve*). Okazuje się, że można tę metodę uogólnić na nasz przypadek. Przyjrzyjmy się temu dokładniej.

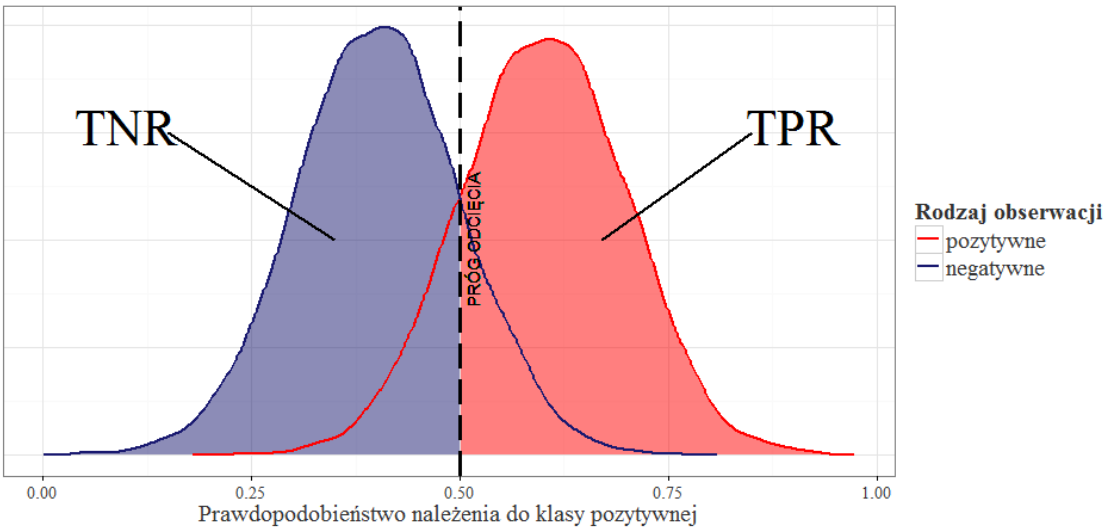
Żeby łatwiej zrozumieć konstrukcję krzywej ROC w przypadku regresji porządkowej, przypomnijmy najpierw, jak otrzymać ją w najprostszym, dwuklasowym przypadku. Załóżmy, że mamy już dopasowany model, a nasza zmienna odpowiedzi jest binarna z odpowiedzią pozytywną lub negatywną. Na zbiorze testowym możemy wtedy otrzymać tabelę jakości dopasowania (patrz Rys.2.1).

		Prawdziwa klasa +	Prawdziwa klasa −
Wyestymowana przez nas klasa	+	TP	FP
	−	FN	TN

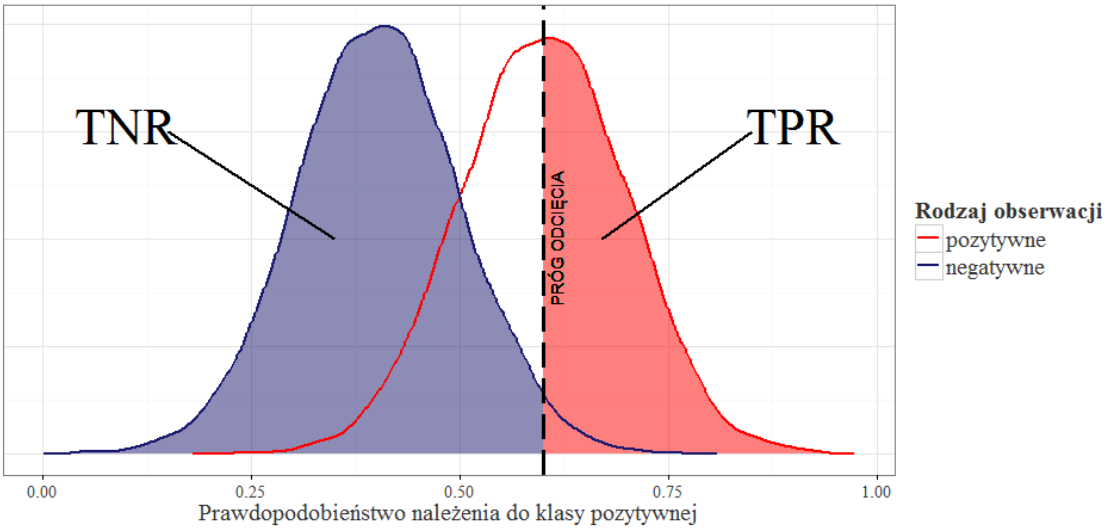
Rysunek 2.1: Tabela jakości dopasowania. TP (ang. *True Positives*) to liczba rekordów z klasy pozytywnej, które zostały zakwalifikowane przez nas jako klasa pozytywna. Analogicznie, TN (ang. *True Negatives*) to liczba rekordów z klasy negatywnej, które zostaly zakwalifikowane przez nas jako klasa negatywna. FP (ang. *False Positives*) oznacza rekordy z klasy negatywnej, zakwalifikowane jako klasa pozytywna i wreszcie, FN (ang. *False Negatives*) to rekordy z klasy pozytywnej, które błędnie zakwalifikowane zostały jako klasa negatywna.



(a) Duże TPR, ale małe TNR.



(b) Zrównoważone TPR i TNR.



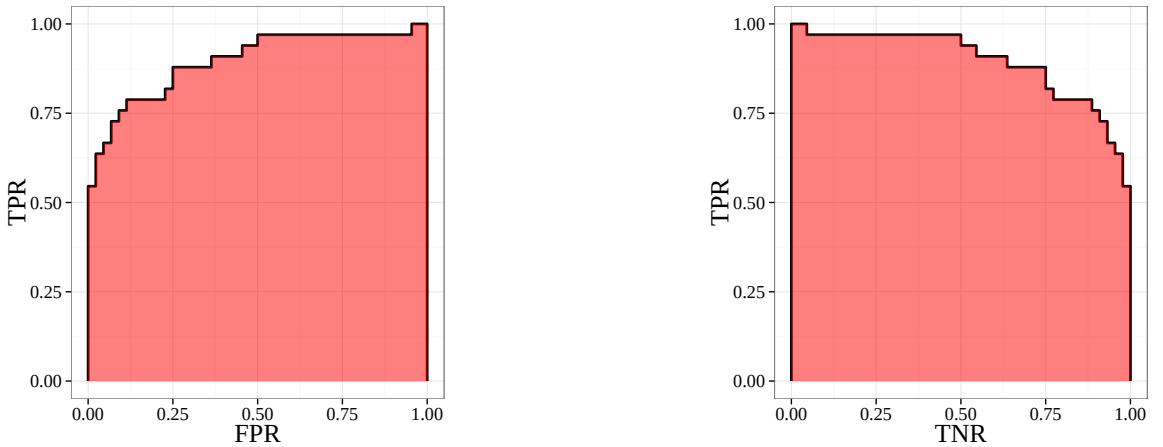
(c) Małe TPR, ale duże TNR.

Rysunek 2.2: Sposób konstruowania krzywej ROC w przypadku dwuklasowym.

Do stworzenia krzywej ROC, potrzebne nam będą dwa wskaźniki – **czułość** (TPR, ang. *True Positive Rate*) i **specyficzność** (TNR, ang. *True Negative Rate*). Czułość definiować będziemy jako prawdopodobieństwo, że pozytywny rekord zostanie poprawnie zakwalifikowany jako pozytywny, a specyficzność jako prawdopodobieństwo, że negatywny rekord zostanie poprawnie zakwalifikowany jako negatywny. Inaczej mówiąc, czułość i specyficzność to procent poprawnie sklasyfikowanych rekordów, odpowiednio w grupie pozytywnej i negatywnej. Korzystając z tabeli jakości dopasowania (patrz Rys. 2.1), otrzymujemy:

$$\begin{aligned} TPR &= \frac{TP}{TP+FN}, \\ TNR &= \frac{TN}{FP+TN}. \end{aligned}$$

Ale jak stworzyć krzywą, mając tylko dwa wskaźniki? Otóż krzywa ROC to wykres punktów (1−TNR, TPR), wyliczonych dla różnych progów odcięcia. Czym jest zatem próg odcięcia? W większości przypadków, model generuje nam nie tylko klasę, do której powinniśmy zaklasyfikować daną obserwację, ale przede wszystkim prawdopodobieństwo, z jakim możemy coś zakwalifikować do klasy pozytywnej. Standardowo przyjmuje się, że to prawdopodobieństwo wynosi $0,5$, ale niekoniecznie musi tak być. Czasem wystarczy nam 40% pewności, żeby coś zaklasyfikować jako pozytywne. Dużo tu zależy od historii, która stoi za naszymi danymi. Na przykład, gdy zajmujemy się klasyfikowaniem pacjentów na chorych i zdrowych, wolimy częściej zakwalifikować kogoś jako chorego, gdy rzeczywiście jest zdrowy, niż odwrotnie, dlatego to prawdopodobieństwo często zmniejszamy. W przypadku analizowania na przykład kampanii reklamowych, może się zdarzyć, że podniesienie progu (czyli zakwalifikowanie mniejszej liczby klientów, jako tych , do których wysłać reklamę) może znacznie zmniejszyć koszty naszej kampanii reklamowej i w rezultacie zwiększyć zysk. I właśnie ten sposób możemy zmieniać próg odcięcia.



(a) Standardowy sposób rysowania krzywej ROC.

(b) Krzywa ROC z TNR (zamiast FPR) na osi OX.

Rysunek 2.3: Przykładowe krzywe ROC.

Przyjrzyjmy się rysunkowi 2.2. Mamy tu wykresy gęstości obserwacji pozytywnych i negatywnych, w zależności od przyjętego progu odcięcia. Większość obserwacji pozytywnych osiąga około 60−procentowe prawdopodobieństwo przynależności do klasy pozytywnej, a negatywnych 40−procentowe prawdopodobieństwo przynależności do klasy negatywnej. Z wykresów wyraźnie widać, że poruszanie tym progiem odcięcia w prawo zwiększy nam czułość, ale

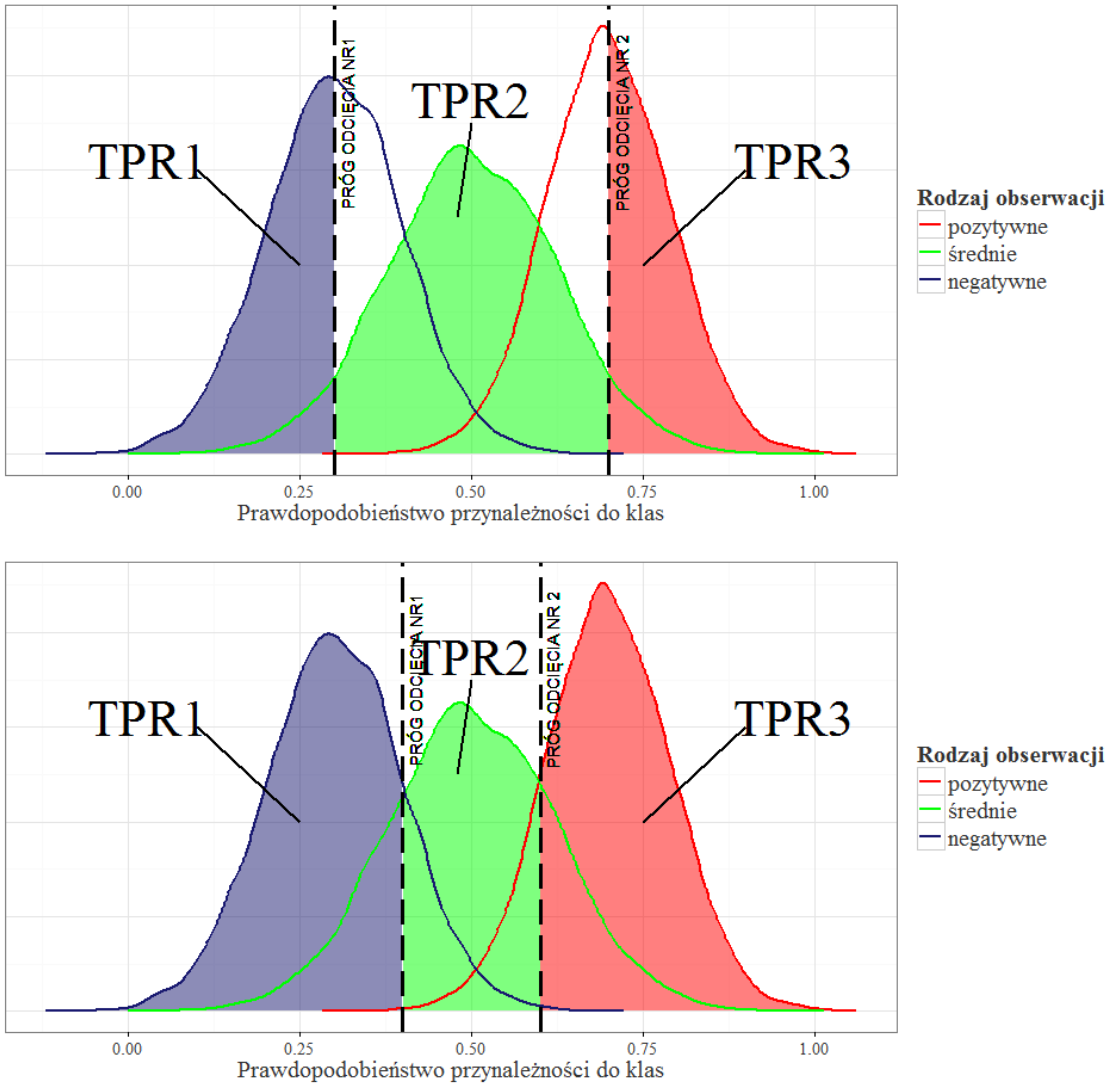
zmniejszy specyficzność, natomiast poruszanie w lewo odwrotnie. Patrząc na krzywą ROC, możemy zobaczyć ich zależność od siebie na jednym wykresie i wybrać taki próg, jaki nam najbardziej odpowiada (najczęściej taki, który jest dobrym kompromisem między czułością a specyficznością). Standardowo, krzywą ROC rysuje się nie w zależności od specyficzności, tylko od jej dopełnienia (tzn. 1−specyficzności), nazywanego FPR (ang. *False Positive Rate*). My jednak, by łatwiej było nam uogólnić krzywą ROC na więcej wymiarów, zastosujemy tę mniej popularną reprezentację, czyli na osi OX będziemy przedstawiać specyficzność (patrz Rys.2.3).

Idealna krzywa to taka, która ma duże TPR i małe FPR, tworzy zatem kwadrat jednostkowy. Zła krzywa, czyli taka, która powstaje, gdy model daje losowe wyniki, to taka, która jest przekątną tego kwadratu. Ponieważ, patrząc na dwie często wielokrotnie przecinające się krzywe ROC, odpowiadające różnym modelom, ciężko jest stwierdzić, która krzywa jest lepsza, wprowadzono współczynnik AUC, czyli pole pod tą krzywą, który pozwala łatwiej to ocenić. Idealny model ma współczynnik AUC równy 1, a model losowy charakteryzuje się AUC równym $0,5$. Na rysunku 2.3 łatwo widać, że w naszym przypadku (czyli z inaczej zdefiniowaną osią OX) współczynnik AUC definiuje się identycznie.

2.4. Krzywa ROC w przypadku regresji porządkowej

W przypadku regresji porządkowej dochodzi problem wielowymiarowości. Przede wszystkim nie mamy tu podziału na klasę pozytywną i negatywną, jak więc stworzyć współczynnik FPR? Można próbować robić to parami tzn. traktować jedną z klas jako pozytywną, a pozostałe połączyć w jedną i traktować jako negatywną. Robiąc w ten sposób z każdą klasą, otrzymamy r (bo tyle jest możliwych poziomów zmiennej odpowiedzi) krzywych ROC, a tym samym r współczynników AUC. Jako ostateczne AUC przyjmuje się wtedy średnią z nich. Nie jest to jednak dobry wskaźnik. Może się bowiem zdarzyć tak, że współczynnik między środkowymi klasami wyjdzie duży, natomiast ten między klasami skrajnymi słaby, tworząc tym samym nienajgorszą średnią. Nie jest to dobre, gdyż często zależy nam na dobrym odróżnieniu właśnie klas skrajnych. Wyobraźmy sobie, że chcemy sprawdzić, czy komuś spodobałaby się sprzedawana przez nas książka. Możliwe odpowiedzi to: bardzo mi się podoba, podoba mi się, nie mam zdania, nie podoba mi się, bardzo mi się nie podoba. Jasne jest, że wolelibyśmy oddzielić klientów, którym bardzo spodobałaby się książka od tych, którym bardzo by się nie spodobała, a nie na przykład tych, którym by się nie spodobała od tych, którym by się bardzo nie spodobała. Żeby udało nam się poradzić sobie z takim problemem, trzeba spojrzeć na niego globalnie.

Opisując krzywą ROC w przypadku dwuklasowym powiedzieliśmy sobie, że będziemy rozważać nie zależność TPR od FPR, ale TPR od TNR. Dlaczego? Właśnie po to, żebyśmy teraz mogli ją łatwiej uogólnić. Zarówno TPR, jak i TNR jest to procent poprawnie sklasyfikowanych odpowiednio pozytywnych bądź negatywnych obserwacji. Nic nie staje zatem na przeszkodzie, by stworzyć r takich współczynników $(\mathrm{TPR}1, \ldots, \mathrm{TPR}r)$, każdy odpowiadający procentowi poprawnie sklasyfikowanych obserwacji z $i-$tej klasy. Przyjmując różne progi odcięcia (patrz Rys. 2.4), których tym razem będzie $r-1$, możemy narysować krzywą ROC, a raczej pewną hiperpowierzchnię. Oczywiście jest to możliwe tylko w przypadku trzyklasowym (patrz Rys. 2.5), ale rysunek taki i tak jest raczej mało czytelny.

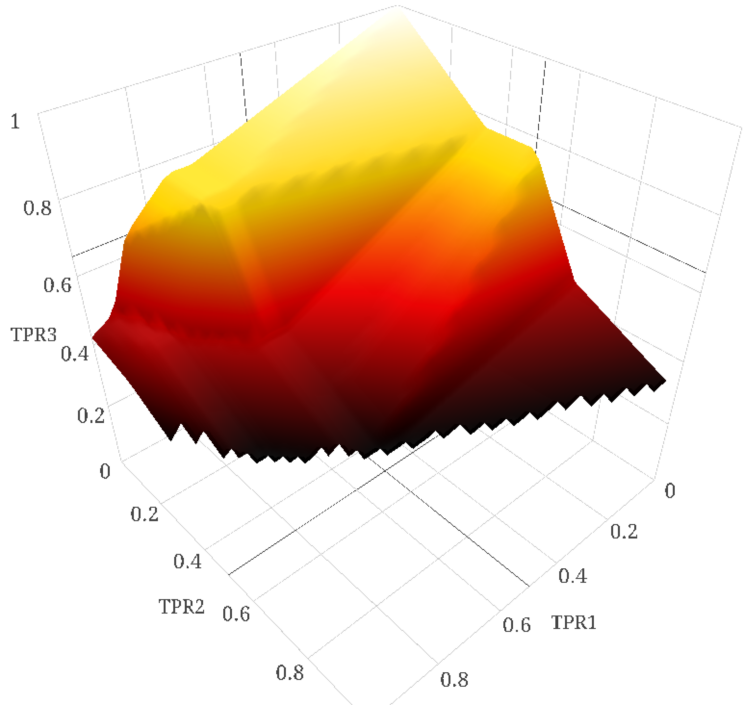


Rysunek 2.4: Sposób konstruowania krzywej ROC w przypadku trzyklasowym.

2.5. Współczynnik VUS

Po co zatem tworzyć wielowymiarową krzywą ROC, skoro i tak trudno cokolwiek z niej odczytać? Otóż głównie po to, by otrzymać współczynnik AUC, który, będąc konkretną liczbą, jest znacznie prostszy w interpretacji. W przypadku więcej niż dwuwymiarowym będziemy go nazywać VUS (ang. *Volume Under the Surface*).

Jako, że liczenie objętości pod hiperpłaszczyzną jest numerycznie raczej trudnym zadaniem, w celu wyliczenia współczynnika VUS, skorzystamy z jego nieco innej interpretacji niż tylko pole pod krzywą ROC. Wróćmy znów do przypadku dwuklasowego i przyjrzyjmy się wykresowi 2.3b. Na osi OY mamy współczynnik TNR, czyli prawdopodobieństwo, że wyestymujemy klasę negatywną pod warunkiem, że klasa rzeczywiście jest negatywna. Równoważnie, można to zapisać jako prawdopodobieństwo, że prawdopodobieństwo odpowiadające negatywnej obserwacji jest mniejsze niż pewien próg odcięcia. Analogicznie TPR to prawdopodobieństwo, że prawdopodobieństwo odpowiadające pozytywnej obserwacji jest większe niż próg odcięcia.



Rysunek 2.5: Krzywa ROC w przypadku trzyklasowym.

Łącząc oba wyniki otrzymamy, że współczynnik AUC to nic innego tylko prawdopodobieństwo, że losowo wybrana pozytywna obserwacja będzie mieć wyższe prawdopodobieństwo niż losowo wybrana negatywna obserwacja. Innymi słowy, będą one dobrze uporządkowane. Łatwo to już uogólnić na więcej wymiarów. Interesować nas będzie pewna estymacja tego prawdopodobieństwa. Łatwo można zauważyć, że będzie nią tzw. statystyka U Manna–Whitney'a–Wilcoxona (por. [10], [11]), czyli wyrażenie:

$$VUS = \frac{1}{n_1 n_2 \cdot \ldots \cdot n_r} \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_r=1}^{n_r} \mathbb{I}_{\{f(\mathbf{x}_{i_1}^1) < \ldots < f(\mathbf{x}_{i_r}^r)\}}, \tag{2.3}$$

gdzie $\mathbf{x}_i^j$ oznacza $i-$ty wektor cech $\mathbf{x}$ o prawdziwej klasie j, n_i to liczba obserwacji zaklasyfikowanych przez nas jako klasa i, a f to pewna funkcja, która zwraca liczbę rzeczywistą, mającą estymować uporządkowanie obserwacji (w większości przypadków, będzie to prawdopodobieństwo zwracane na koniec).

Krzywa ROC i współczynnik VUS jest więc dość prostym, bardzo łatwo interpretowalnym i pomocnym narzędziem do oceny jakości modelu i podejmowania decyzji, który model jest najlepszy. Największą jego wadą wydaje się konieczność znania prawdopodobieństw przynależności do klas (lub po prostu funkcji, która pozwoli nasze obserwacje uporządkować), a nie każda metoda takie prawdopodobieństwa zwraca (np. nie robią tego sieci neuronowe). Trzeba wtedy odwołać się do prostszych metod (takich jak procent poprawności dopasowania lub czułość). Większość modeli oferuje jednak taką możliwość, więc niewątpliwie warto z tego narzędzia diagnostycznego korzystać.

Rozdział 3

Symulacje

W poprzednich rozdziałach poznaliśmy różne techniki radzenia sobie z problemem regresji porządkowej. Znamy już również techniki oceny klasyfikatorów. Spróbujmy teraz zobaczyć, jak na rzeczywistych danych zachowuje się każda z poznanych metod.

3.1. Opis danych

Do dyspozycji mamy dziewięć zbiorów (*abalone, auto, diabetes, housing, machine, pyrim, stock, triazines* oraz *wpbc*) o różnej liczbie klas (5 lub 10), różnej liczbie obserwacji oraz różnym rozkładzie klas (patrz Rys. 3.1). Zbiory znaleźć można tu: [12].

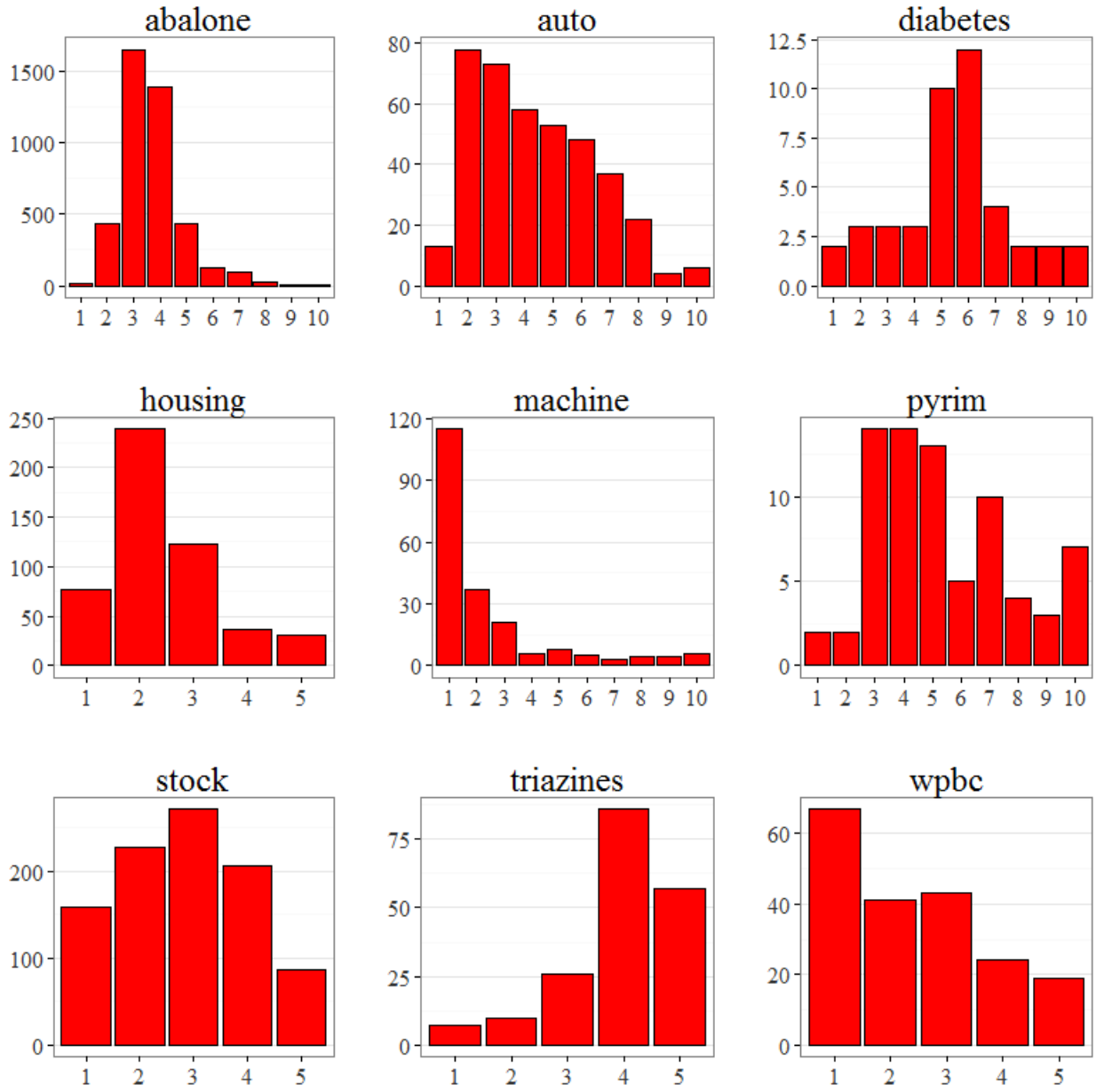
Każdy ze zbiorów został losowo podzielony na zbiór treningowy (do budowy modelu) i zbiór testowy (do oceny jakości modelu) w proporcji $7:3$. Następnie na każdym zbiorze treningowym zbudowany został każdy z pięciu modeli (model proporcjonalnych szans, model oparty o procesy gaussowskie, model Franka i Halla, sieci neuronowe oraz SVM) i przetestowany na zbiorze treningowym. Dla każdego zbioru i każdej metody otrzymaliśmy współczynniki VUS, PPK i ABSerr (por. równania 2.1, 2.2, 2.3). Wszystkie wyniki zobaczyć można w tabeli 3.1.

3.2. Analiza wyników

Pierwszą rzeczą, która rzuca się w oczy, patrząc na tabelę wyników (Tabela 3.1), jest niska jakość współczynnika VUS. W większości przypadków jest on zerowy lub bardzo mały. Przypomnijmy, że reprezentuje on pewne prawdopodobieństwo, wobec czego już nawet wynik 50% byłby nienajlepszy, a co dopiero 0%. Sensowny wynik współczynnika VUS zobaczyć możemy tak naprawdę jedynie w przypadku zbiorów *housing* i *stock*.

Kolejną rzeczą jest niespodziewanie dobre zachowanie sieci neuronowych. Osiągają one najlepszy procent poprawnej klasyfikacji w 56% zbiorów, a w 22% niewiele odbiegają od najlepszego wyniku. Niestety nie możemy tutaj porównać współczynnika VUS.

No i w końcu, widać wyraźnie, że nie ma jednej najlepszej metody. Każda analizowana metoda „wygrywa" na którymś ze zbiorów.



Rysunek 3.1: Rozkłady odpowiedzi poszczególnych zbiorów danych.

Analizując powyższe wyniki, nasuwają się na myśl ciekawe pytania. Przede wszystkim dlaczego współczynnik VUS wychodzi taki słaby? Czy może zależy to od liczby klas zmiennej odpowiedzi? Czy każda z metod też od tego zależy? W następnym podrozdziale spróbujemy sobie na te pytania odpowiedzieć.

3.3. Dane o różnej liczbie klas

Weźmy pod uwagę zbiór danych *machine*. Zmienna odpowiedzi jest tu zmienną ciągłą, która w powyższej analizie została – w sposób lekko szutuczny – zdyskretyzowana. Zróbmy teraz to samo, ale dyskretyzując zbiór na, odpowiednio, 3, 5, 7, i 10 klas. By sprawdzić dodatkowo,

		Procesy gaussowskie	Model proporcjonalnych szans	Sieci neuronowe	Metoda Franka i Halla	Wektory maszyn podpierających (SVM)
abalone	VUS [%]	0,00	0,00	–	0,00	0,00
	PPK [%]	56,38	55,18	56,86	55,18	56,30
	ABSerr	0,55	0,55	0,51	0,55	0,52
auto	VUS [%]	3,54	1,13	–	0,25	0,81
	PPK [%]	37,29	49,15	48,31	43,22	38,98
	ABSerr	1,08	0,63	0,66	0,73	0,90
diabetes	VUS [%]	5,00	5,00	–	15,00	5,00
	PPK [%]	23,08	23,08	23,08	15,38	23,08
	ABSerr	1,46	1,31	1,23	1,23	1,38
housing	VUS [%]	54,46	55,31	–	42,06	51,33
	PPK [%]	67,11	67,11	75,00	72,37	72,37
	ABSerr	0,36	0,35	0,26	0,28	0,29
machine	VUS [%]	0,00	0,00	–	0,00	0,00
	PPK [%]	57,14	68,25	66,67	66,67	60,32
	ABSerr	1,17	0,52	0,51	0,67	0,73
pyrim	VUS [%]	1,11	0,00	–	0,56	0,93
	PPK [%]	21,74	8,70	52,17	30,43	47,83
	ABSerr	1,30	1,65	0,91	0,91	0,96
stock	VUS [%]	53,97	73,12	–	56,53	96,01
	PPK [%]	58,95	72,63	83,16	81,40	91,58
	ABSerr	0,42	0,28	0,18	0,19	0,08
triazines	VUS [%]	1,84	1,83	–	5,45	1,96
	PPK [%]	51,79	39,29	39,29	53,57	51,79
	ABSerr	0,64	0,91	0,71	0,61	0,64
wpbc	VUS [%]	4,44	3,99	–	0,63	1,88
	PPK [%]	33,90	30,51	35,59	28,81	28,81
	ABSerr	1,12	1,25	0,86	1,02	0,92

Tabela 3.1: Tabela wyników. Na czerwono zaznaczony jest najlepszy wynik dla każdego wskaźnika każdego zbioru. Dla sieci neuronowych nie da się niestety obliczyć współczynnika VUS.

czy różnice zależą od liczby klas, czy może jednak od ich liczności, rozważać będziemy dwa rodzaje dyskretyzacji: równomierną (tzn. taką, która do każdej klasy przydziela tyle samo obserwacji) i klastrową (tzn. taką, która szuka w danych naturalnych klastrów).

	Procesy gaussowskie			Sieci neuronowe			Model proporcjonalnych szans			Metoda Franka i Halla			Wektory maszyn podpierających (SVM)		
Liczba klastrów	VUS	PPK	ABSerr	VUS	PPK	ABSerr	VUS	PPK	ABSerr	VUS	PPK	ABSerr	VUS	PPK	ABSerr
3	81,55	66,67	0,33	–	71,43	0,29	97,62	92,86	0,07	36,67	88,1	0,12	65,77	61,9	0,38
5	22,85	21,43	1,88	–	52,38	0,52	50,75	76,19	0,26	0,08	76,19	0,36	12,79	38,1	0,76
7	9,38	38,1	1,1	–	42,86	0,6	35,94	54,76	0,48	0	61,9	0,52	1,88	16,67	1,31
10	0,7	33,33	1,88	–	35,71	0,9	5,1	50	0,64	0	45,24	0,95	0,01	19,05	1,62

Tabela 3.2: Wyniki analizy zbioru *machine* dla różnej liczby klas zmiennej odpowiedzi, stosując **dyskretyzację klastrową**.

	Procesy gaussowskie			Sieci neuronowe			Model proporcjonalnych szans			Metoda Franka i Halla			Wektory maszyn podpierających (SVM)		
Liczba klastrów	VUS	PPK	ABSerr	VUS	PPK	ABSerr	VUS	PPK	ABSerr	VUS	PPK	ABSerr	VUS	PPK	ABSerr
3	0,48	83,33	0,19	–	95,24	0,05	81,62	71,43	0,29	41,52	64,29	0,36	91,43	88,1	0,12
5	28,83	2,38	3,43	–	80,95	0,21	35,97	64,29	0,38	10,11	47,62	0,57	32,33	76,19	0,29
7	0	19,05	1,81	–	54,76	0,48	8,54	45,24	0,64	0,16	33,33	0,86	6,89	30,95	0,86
10	0,45	2,38	5,45	–	47,62	0,64	0,8	38,1	0,88	0	23,81	1,38	0,05	30,95	1,14

Tabela 3.3: Wyniki analizy zbioru *machine* dla różnej liczby klas zmiennej odpowiedzi, stosując **dyskretyzację równomierną**.

Przyjrzyjmy się tabelom 3.2 i 3.3. Widać wyraźnie, że – niezależnie od sposobu dyskretyzacji danych – wraz ze wzrostem liczby klas, dopasowanie modelu maleje. O ile jednak współczynnik PPK zachowuje się w miarę sensownie nawet aż przy dziesięciu klasach, o tyle VUS przy tej samej liczbie klas, staje się całkowicie bezużyteczny. Niemniej jednak, w przypadku, gdy współczynnik VUS jest w miarę sensowny, należy wziąć go pod uwagę. Czasami warto zgodzić się na niższy PPK, żeby zyskać wyższy VUS.

Zastanówmy się teraz, jaki wpływ na poszczególne metody ma równoliczność klas. Widać wyraźnie, że metodami, które odniosły duże korzyści ze zbalansowania zbioru danych są sieci neuronowe i SVM-y. W metodzie Franka i Halla znacząco polepszył się współczynnik VUS, jednak spadł procent poprawnej klasyfikacji. Przypatrując się wynikom procesów gaussowskich i modelowi proporcjonalnych szans, wygląda na to, że metody mające duże założenia co do rozkładów (normalność w procesie gaussowskim i założenie proporcjonalnych szans) nie działają najlepiej dla sztucznie wprowadzonych (mimo, że równolicznych) klas. Wskazuje to na wagę tych założeń.

Dodatek A

Wyprowadzenia pomocniczych twierdzeń

A.1. Wzór Bayesa dla więcej niż jednego warunku

Korzystając ze wzoru Bayesa dla prawdopodobieństwa warunkowego i rozpisując jedynie warunek B, a warunek A pozostawiając bez zmian, otrzymujemy:

$$\mathbb{P}(A|B,C) = \frac{\mathbb{P}(B|A,C)\mathbb{P}(A|C)}{\mathbb{P}(B|C)}.$$

Następnie, zakładając, że A jest zależne od C, możemy w prawdopodobieństwie $\mathbb{P}(B|A,C)$ pominąć warunek C, gdyż jest on już niejako zawarty w warunku A. W rezultacie otrzymujemy:

$$\mathbb{P}(A|B,C) = \frac{\mathbb{P}(B|A)\mathbb{P}(A|C)}{\mathbb{P}(B|C)}.$$

A.2. Całka z iloczynu dystrybuanty i gęstości rozkładu normalnego

Niech $X \sim \mathcal{N}(\mu, \sigma^2)$ o gęstości $f(\cdot)$, zaś Φ to dystrybuanta standardowego rozkładu normalnego (czyli $\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(y)dy$). Interesuje nas policzenie całki

$$I := \int_{\mathbb{R}} \Phi(\frac{x-m}{\nu}) f(x)dx.$$

Zacznijmy od zwykłego rozpisania podstawowych symboli.

$$I = \int_{\mathbb{R}} \int_{-\infty}^{\frac{x-m}{\nu}} \frac{1}{\sqrt{2\Pi}} e^{-\frac{y^2}{2}} dy \cdot \frac{1}{\sqrt{2\Pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \frac{1}{2\Pi\sigma} \int_{\mathbb{R}} \int_{-\infty}^{\frac{x-m}{\nu}} e^{-\frac{y^2}{2} - \frac{(x-\mu)^2}{2\sigma^2}} dydx$$

Następnie zróbmy po kolei trzy podstawienia $u = \nu y + m$, $z = (u-m) - (x-\mu)$ oraz $w = x - \mu$ i zamieńmy kolejność całkowania.

$$\begin{aligned} I &= \frac{1}{2\Pi\sigma\nu}\int_{\mathbb{R}}\int_{-\infty}^{x} e^{-\frac{(u-m)^2}{2\nu^2}-\frac{(x-\mu)^2}{2\sigma^2}}dudx = \frac{1}{2\Pi\sigma\nu}\int_{\mathbb{R}}\int_{-\infty}^{\mu-m} e^{-\frac{(z+(x-\mu))^2}{2\nu^2}-\frac{(x-\mu)^2}{2\sigma^2}}dzdx = \\ &= \frac{1}{2\Pi\sigma\nu}\int_{-\infty}^{\mu-m}\int_{\mathbb{R}} e^{-\frac{(z+w)^2}{2\nu^2}-\frac{w^2}{2\sigma^2}}dwdz \end{aligned}$$

Zajmijmy się na razie tylko środkową całką. Po sprowadzeniu do wspólnego mianownika i prostych przekształceniach, otrzymamy:

$$\begin{aligned} A &= \int_{\mathbb{R}} e^{-\frac{(z+w)^2}{2\nu^2}-\frac{w^2}{2\sigma^2}}dw = \int_{\mathbb{R}} e^{-\frac{\left(w\sqrt{\sigma^2+\nu^2}+z\frac{\sigma^2}{\sqrt{\sigma^2+\nu^2}}\right)^2}{2\nu^2\mu^2}-\frac{z^2}{2(\sigma^2+\nu^2)}}dw = \\ &= e^{-\frac{z^2}{2(\sigma^2+\nu^2)}}\int_{\mathbb{R}} e^{-\frac{\left(w\sqrt{\sigma^2+\nu^2}+z\frac{\sigma^2}{\sqrt{\sigma^2+\nu^2}}\right)^2}{2\nu^2\mu^2}}dw. \end{aligned}$$

Robiąc podstawienie $u = \frac{w\sqrt{\sigma^2+\nu^2}+z\frac{\sigma^2}{\sqrt{\sigma^2+\nu^2}}}{\nu\sigma}$ oraz korzystając z faktu, że gęstość rozkładu prawdopodobieństwa całkuje się do jedynki, otrzymamy:

$$A = e^{-\frac{z^2}{2}}\cdot\sqrt{2\Pi}\cdot\underbrace{\frac{1}{\sqrt{2\Pi}}\int_{\mathbb{R}} e^{-\frac{u^2}{2}}du}_{=1}\cdot\frac{\nu\sigma\sqrt{\sigma^2+\nu^2}}{\sigma^2+\nu^2} = \sqrt{2\Pi}\frac{\nu\sigma}{\sqrt{\sigma^2+\nu^2}}e^{-\frac{z^2}{2(\sigma^2+\nu^2)}}.$$

Wróćmy teraz do szukanej całki.

$$I = \frac{1}{2\Pi\sigma\nu}\int_{-\infty}^{\mu-m} A\ \ dz = \frac{1}{2\Pi\sigma\nu}\sqrt{2\Pi}\frac{\nu\sigma}{\sqrt{\sigma^2+\nu^2}}\int_{-\infty}^{\mu-m} e^{-\frac{z^2}{2(\sigma^2+\nu^2)}}dz$$

Robiąc podstawienie $x = \frac{z}{\sqrt{\sigma^2+\nu^2}}$, otrzymamy:

$$I = \frac{1}{\sqrt{2\Pi}}\int_{-\infty}^{\frac{\mu-m}{\sqrt{\sigma^2+\nu^2}}} e^{-\frac{x^2}{2}}dx = \Phi\left(\frac{\mu-m}{\sqrt{\sigma^2+\nu^2}}\right).$$

Literatura

[1] Dobson A. J., An Introduction to Generalized Linear Models, 2nd Edition, 2001.

[2] Frank E., Hall M., A simple approach to ordinal classification, *Proceedings of the European Conference on Machine Learning*, Freibourg, Niemcy, 2001, str. 146–156.

[3] Cheng J., Wang Z., Pollastri G., A neural network approach to ordinal regression, *Neutral Networks*, Hong Kong, 2008.

[4] Koronacki J., Ćwik J., Statystyczne systemy uczące się, Warszawa, 2005.

[5] Chu W., Sathiya Keerthi S., *Support Vector Ordinal Regression*

[6] Chu W., Ghahramani Z., *Gaussian Processes for Ordinal Regression.*

[7] Ebden M., *Gaussian Processes for Classification: A Quick Introduction*, August 2008.

[8] Ebden M., *Gaussian Processes for Regression: A Quick Introduction*, August 2008.

[9] Rasmussen C., Williams C., *Gaussian Processes for Machine Learning*, 2006.

[10] Waegman W., De Baets B., A survey on ROC-based ordinal regression, w: Fürnkranz J., Hüllermeier E. (Eds.), *Preference Learning*, Springer, 2010, str. 127-154.

[11] Nakas C.T., Yiannoutsos C.T., Ordered Multiple Class Receiver Operating Characteristic (ROC) Analysis, *Encyclopedia of Biopharmaceutical Statistics*, Taylor and Francis, 2006.

[12] Chu W., Benchmark of ordinal regression, `http://www.gatsby.ucl.ac.uk/~chuwei/ordinalregression.html`

[13] Torgo L., Regression Data Sets, `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`

[14] Cheng J., Department of Computer Science, University of Missouri, `http://sysbio.rnet.missouri.edu/multicom_toolbox/nnrank%201.1.html`

[15] Chu W., Source Code for Support Vector Ordinal Regression, `http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm`

[16] Chu W., Source Code for Gaussian Processes for Ordinal Regression, `http://www.gatsby.ucl.ac.uk/~chuwei/README.gpor`

Marta Sommer
Nr albumu 237503

Warszawa, 27 października 2015

Oświadczenie

Oświadczam, że pracę magisterską pod tytułem „Statystyczne metody regresji porządkowej", której promotorem jest prof. nzw. dr hab. Przemysław Grzegorzewski wykonałam samodzielnie, co poświadczam własnoręcznym podpisem.

.....
Marta Sommer