



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

<b>Wstęp</b>	<b>5</b>
<b>1. Opis teoretyczny dostępnych metod</b>	<b>7</b>
1.1. Postawienie problemu i podstawowe oznaczenia . . . . .	7
1.2. Model proporcjonalnych szans . . . . .	7
1.3. Wektory maszyn podpierających (SVM) . . . . .	8
1.4. Sieci neuronowe . . . . .	9
1.5. Metoda zaproponowana przez E. Franka i M. Halla . . . . .	11
1.6. Procesy gaussowskie . . . . .	13
<b>A. ahdgkhakf</b>	<b>17</b>
<b>Literatura</b>	<b>19</b>



# 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  $\mathbf{x} = (x_1, \dots, x_d)$ , 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  $\mathbf{y} = (y_1, \dots, y_r)$ , 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  $\mathcal{Y} = \{1, \dots, r\}$  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. Jest to jedna z metod uogólnionych modeli liniowych, bardzo silnie opierająca się na regresji logistycznej. Interesują nas prawdopodobieństwa:

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

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

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

Następnie rozważa się poniższy model logitowy:

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

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

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

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

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 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. W dużym uproszczeniu, polega ona na konstrukcji dwóch równoległych i maksymalnie oddalonych od siebie hiperpłaszczyzn w taki sposób, by minimalizować kwadrat odległości między nimi. By móc obsługiwać przypadki, w których brak liniowej separowalności, wprowadza się dodatkowo karę za nieidealne rozdzielenie klas. W przypadku dwuklasowym budowa modelu sprowadza się do rozwiązania następującego problemu optymalizacyjnego:

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

przy ograniczeniach:

$$\begin{cases} \mathbf{x}^{(i)T} \mathbf{w} + b & \geq 1 - \xi_i \\ \mathbf{x}^{(i)T} \mathbf{w} + b & \leq -1 + \xi_i, \end{cases}$$

gdzie  $\mathbf{w} \in \mathbb{R}^K$ ,  $b \in \mathbb{R}$ ,  $C \in \mathbb{R}$  i  $\xi_i \geq 0$  są parametrami modelu.

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

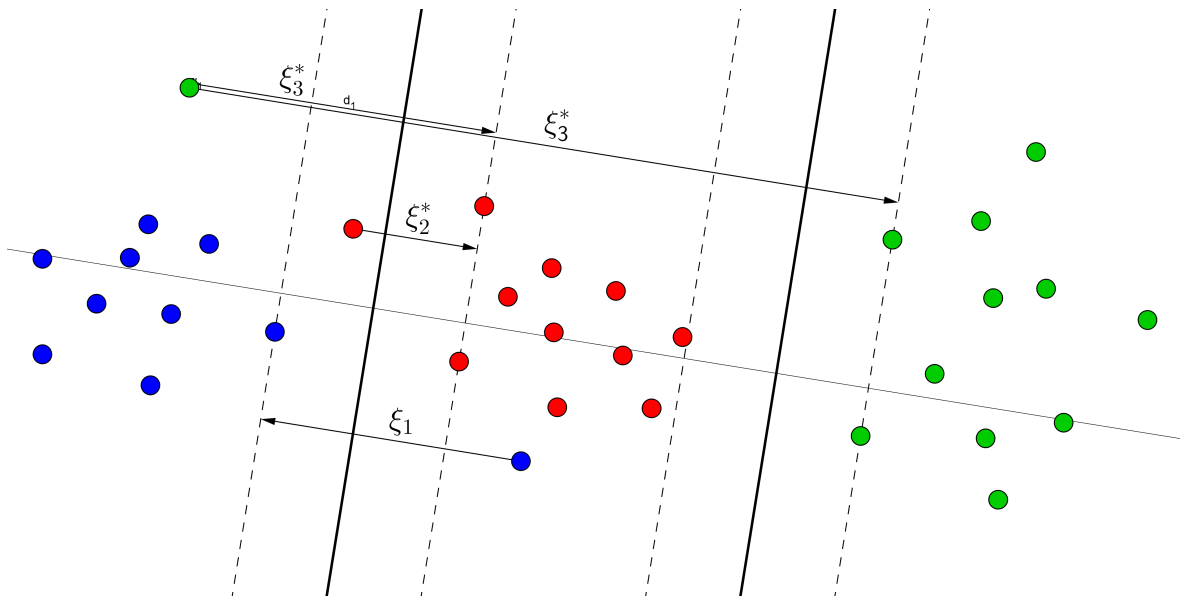
Tym razem optymalizować będziemy następujące wyrażenie:

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

przy ograniczeniach:

$$\begin{cases} \mathbf{w}x_k^{(i)} - b_j & \leq -1 + \xi_{ki}^j, & \text{dla } k = 1, \dots, j \text{ oraz } i = 1, \dots, n_k \\ \mathbf{w}x_k^{(i)} - b_j & \geq +1 - \xi_{ki}^{*j}, & \text{dla } k = j + 1, \dots, r \text{ oraz } i = 1, \dots, n_k \end{cases}$$





Rysunek 1.1: Przykładowa klasyfikacja metodą SVM.

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. Dla progu  $b_j$ , wartości funkcji  $\mathbf{w}x_k^{(i)}$  dla obserwacji z niższych klas, powinny być niższe niż dolna granica  $b_j - 1$ . Jeśli tak nie jest, wtedy jako błąd próbki  $\xi_{ki}^j$  dla progu  $b_j$  uznaje się  $\xi_{ki}^j = \mathbf{w}x_k^{(i)} - (b_j - 1)$ . Analogicznie, dla obserwacji z wyższych klas, otrzymujemy błędy  $\xi_{ki}^{*j}$ .

Budowa modelu i tym razem sprowadza się więc do problemu optymalizacyjnego.

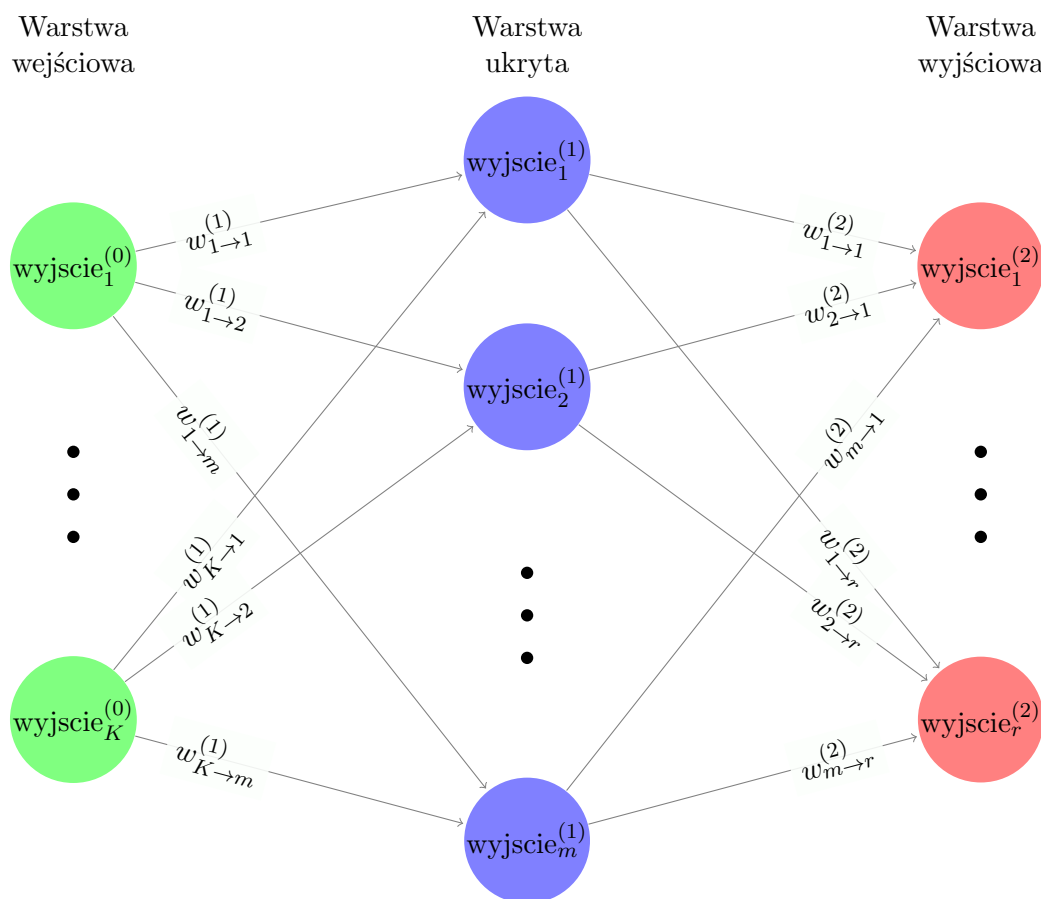
## 1.4. Sieci neuronowe

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. [2]).

Standardowo, na wejściu otrzymujemy zbiór uczący w postaci  $n$  par  $(\mathbf{x}, y)$ , gdzie  $\mathbf{x} = (x_1, \dots, x_K)^T$  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  $\mathbf{y} = (y_1, \dots, y_r)^T$  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  $\mathbf{y} = (0, 0, \dots, 1, \dots, 0)^T$ , mającego jedynkę na  $i$ -tym miejscu, jeśli obserwacja należała do  $i$ -tej klasy, rozważać będziemy wektor  $\mathbf{y} = (1, 1, \dots, 1, \dots, 0)^T$ , 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 przyjmujemy funkcję sigmoidalną  $f(x) = \frac{1}{1+e^{-x}}$ , dobrze reprezentującą przynależność do danej klasy jako prawdopodobieństwo. Może się zdarzyć, że wyjściowy wektor nie będzie ciągiem malejącym (co trochę przeczy intuicji), jednak nie jest to konieczne do robienia predykcji.



Rysunek 1.2: Przykładowa sieć neuronowa.

Uczenie sieci neuronowej będzie się odbywało algorytmem propagacji wstecznej z kwadratową funkcją straty (można też użyć jakiejś innej np. entropii). Algorytm wygląda następująco:

1. Wybieramy małe wagi początkowe oraz 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).

Ad. 3)

Dla każdego neuronu obliczamy wartość wejściową ze wzoru:

$$wejście_j^{(i)} = \sum_{k: \exists w_{k \rightarrow j}^{(i)}} \left( w_{k \rightarrow j}^{(i)} \cdot wyjście_k^{(i-1)} \right),$$

gdzie  $wyjście_i^{(0)} = x_i$ . A następnie wyjściową:

$$wyjście_j^{(i)} = f \left( wyjście_j^{(i)} \right).$$

Ad. 4)

Dla warstwy wyjściowej błąd ma postać:

$$\delta_j = -2 \cdot wyjście_j^2 \cdot (1 - wyjście_j)^2 \cdot (y_j - wyjście_j),$$

zaś dla warstw ukrytych:

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

Ad. 5)

Modyfikacja wag przebiega następująco:

$$w_{k \rightarrow j}^{(i)(new)} = w_{k \rightarrow j}^{(i)(old)} - \eta \cdot \delta_j^{(i)} \cdot wyjście_k^{(i-1)}.$$

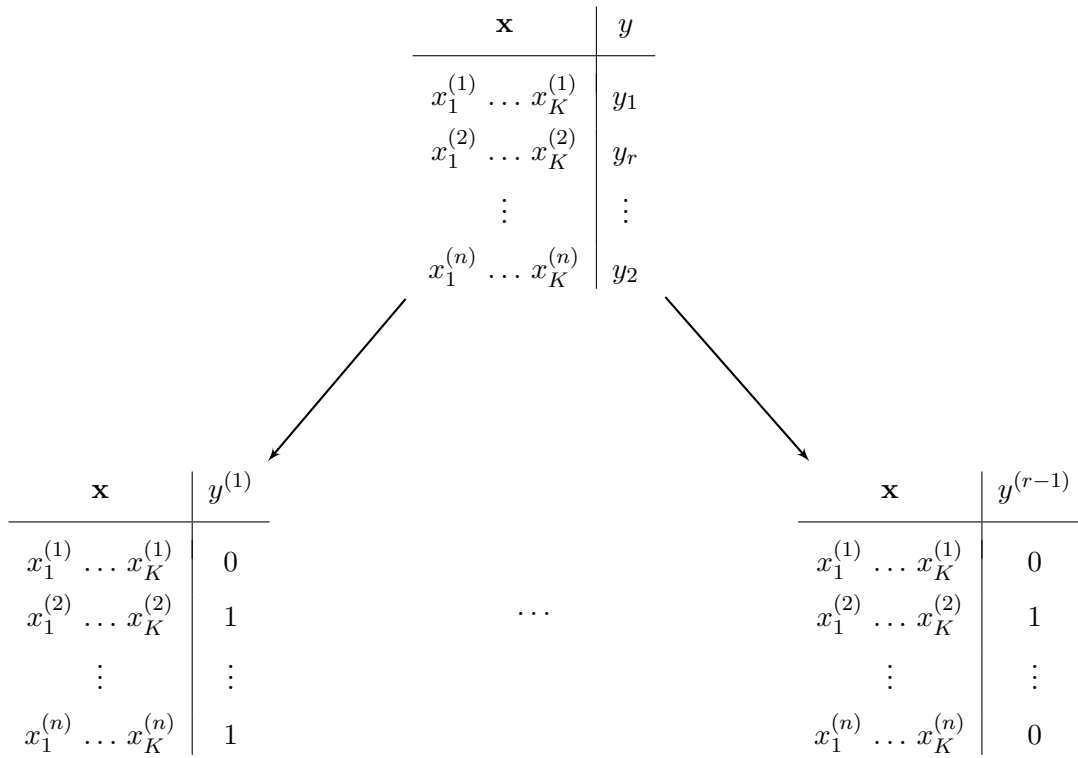
Predykcja opiera się już tylko na przejściu algorytmu w przód z nowymi obserwacjami wejściowymi  $\mathbf{x}_*$  i ustaleniu progu (najczęściej równego 0,5), klasyfikującego neuron wyjściowy jako jedynekę. Skanujemy wektor wyjściowy zaczynając od  $y_1$  i kończymy, gdy pierwszy raz natkniemy się na 0. Przypisujemy obserwacji taką klasę, jaką długość miał znaleziony przez nas ciąg jedynek.

## 1.5. Metoda zaproponowana przez E. Franka i M. Halla

Podejście Franka i Halla (por. [1]) do zagadnienia regresji porządkowej jest nieco inne, niż przedstawione do tej pory metody. Opiera się bowiem nie na stworzeniu nowego modelu, ale na odpowiednim przedefiniowaniu zbioru danych, a następnie na sprowadzeniu zadania do problemu zwykłej klasyfikacji z dwoma klasami. Przekształcamy zatem  $r$ -klasowy model regresji porządkowej do  $(r - 1)$  dwuklasowych problemów klasyfikacji.

Algorytm do 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) taki, który zwraca prawdopodobieństwa przynależności do klas.
3. Robimy predykcję dla nowej obserwacji.



Rysunek 1.3: Modyfikacja przykładowego zbioru uczącego.

Ad. 1)

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

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

Ad. 3)

Dla nowego wektora atrybutów  $\mathbf{x}$  robimy predykcję na  $r - 1$  modelach uzyskanych w punkcie 2). Zwracamy jednak nie predykcję klasy, ale prawdopodobieństwo przynależności do klasy pierwszej. Uzyskujemy w ten sposób  $r - 1$  następujących prawdopodobieństw:

$$\begin{aligned}
 \mathbb{P}(y > 1) \\
 \vdots \\
 \mathbb{P}(y > r - 1).
 \end{aligned}$$

Nas natomiast interesują prawdopodobieństwa:

$$\begin{aligned}\mathbb{P}(y &= 1) \\ &\vdots \\ \mathbb{P}(y &= r-1).\end{aligned}$$

Łatwo otrzymamy korzystając z następującego wzoru łańcuchowego:

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

Ostatecznie, nowej obserwacji przypisujemy 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 szczególnie przy zwykłej regresji, znalazła ona jednak również zastosowanie w klasyfikacji zarówno jedno, jak i wieloetykietowej. Chu i Ghahramani w pracy [5] 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. W celu wyrobienia sobie intuicji, przeanalizujemy całe rozumowanie nieco 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 charakterystykę takiego procesu tworzą dwie informacje – średnia (standardowo przyjmuje się 0) oraz macierz kowariancji  $\Sigma$ . Dla celów tej pracy przyjmujemy, że elementy macierzy kowariancji definiowane są w następujący sposób:

$$\Sigma_{ij} = \Sigma(x_i, x_j) = \exp \left\{ -\frac{\kappa}{2} \sum_{\xi=1}^K (x_i^\xi - x_j^\xi)^2 \right\},$$

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

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

gdzie  $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^T$  to wektor zawierający realizację zmiennej ukrytej, odpowiadającą 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. Przy pomocy zmiennej ukrytej rozpiszmy je w następujący sposób:

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

Analogicznie:

$$\mathbb{P}(f(\mathbf{x}_*)|\mathbf{X}, \mathbf{y}) = \int \mathbb{P}(f(\mathbf{x}_*)|\mathbf{f})\mathbb{P}(\mathbf{f}|\mathbf{X}, \mathbf{y})d\mathbf{f}. \quad (1.2)$$

Szukamy więc powyższych prawdopodobieństw, by następnie całość scałkować.

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

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

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

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

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

$$\mathbb{P}(\mathbf{f}|\mathbf{X}, \mathbf{y}) = \frac{\mathbb{P}(\mathbf{y}|\mathbf{f})\mathbb{P}(\mathbf{f}|\mathbf{X})}{\mathbb{P}(\mathbf{y}|\mathbf{X})}. \quad (1.3)$$

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. wiarygodność. Ponieważ wszystkie obserwacje są niezależne, otrzymujemy:

$$\mathbb{P}(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n \mathbb{P}(y_i|f(\mathbf{x}_i)). \quad (1.4)$$

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  $i = 1, \dots, r-1$  to parametry modelu. Wygodniej, można  $b_i$  sparametryzować jako:  $b_1 \in \mathbb{R}$  oraz  $b_i = \sum_{t=2}^i \Delta_t + b_1$ , gdzie  $\Delta_t > 0$  oraz  $j = 2, \dots, r-1$ . Bardzo rzadko mamy jednak do czynienia z sytuacją idealną, dlatego będziemy budować model, zakładając dodatkowy szum  $\delta$  o rozkładzie  $\mathcal{N}(0, \sigma^2)$ . Wtedy prawdopodobieństwo zmienia się następująco:

$$\begin{aligned} \mathbb{P}(y_i|f(\mathbf{x}_i)) &= \int \mathbb{P}_{ideal}(y_i, \delta_i|f(\mathbf{x}_i))d\delta_i = \int \mathbb{P}_{ideal}(y_i|f(\mathbf{x}_i), \delta_i)\mathbb{P}(\delta_i)d\delta_i = \\ &= \int \mathbb{P}(\delta_i)\mathbb{I}_{\{f(\mathbf{x}_i) + \delta_i \in (b_{y_i-1}, b_{y_i}]\}}d\delta_i = \Phi\left(\frac{b_{y_i} - f(x_i)}{\sigma}\right) - \Phi\left(\frac{b_{y_i-1} - f(x_i)}{\sigma}\right), \end{aligned}$$

gdzie  $\Phi(\cdot)$  to dystrybucja 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 mamy, że:

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

Korzystając z równań (1.1), (1.3) i (1.4) można łatwo zobaczyć, że:

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

gdzie  $\ell(y_i, f(\mathbf{x}_i)) := -\ln \mathbb{P}(y_i|f(\mathbf{x}_i))$ . Nie da się znaleźć minimum tej funkcji analitycznie, ponieważ  $\ell(y_i, f(\mathbf{x}_i))$  nie jest gaussowska. 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 (1.2), 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

$$\boldsymbol{\Lambda} = \begin{bmatrix} \frac{\partial^2 \ell(y_1, f(\mathbf{x}_1))}{\partial^2 f(\mathbf{x}_1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial^2 \ell(y_n, f(\mathbf{x}_n))}{\partial^2 f(\mathbf{x}_n)} \end{bmatrix}.$$

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

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

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

$$\mathbf{f}|\mathbf{X}, \mathbf{y} \sim \mathcal{N}(\hat{\mathbf{f}}, (\boldsymbol{\Sigma}^{-1} + \hat{\boldsymbol{\Lambda}})^{-1}).$$

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

$$\mathbb{P}(\mathbf{y}|\mathbf{X}) = \dots$$





**Dodatek A**

**ahdgkhakf**



# Literatura

- [1] Frank E., Hall M., A simple approach to ordinal classification, *Proceedings of the European Conference on Machine Learning*, Freiburg, Niemcy, 2001, str. 146–156.
- [2] Cheng J., Wang Z., Pollastri G., A neural network approach to ordinal regression, *Neural Networks*, Hong Kong, 2008.
- [3] Dobson A. J., An Introduction to Generalized Linear Models, 2nd Edition, 2001
- [4] Chu W., Sathya Keerthi S., *Support Vector Ordinal Regression*
- [5] Chu W., Ghahramani Z., *Gaussian Processes for Ordinal Regression*



Marta Sommer  
Nr albumu 237503

Warszawa, 1 lipca 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