Spis treści

1 Wstęp

Celem tych notatek jest zwięzłe przedstawienie kompletu zagadnień związanych z szeroko pojętym uczeniem głębokim jako podejściem do Sztucznej Inteligencji (SI). Zaczynamy od minimalnego zbioru wymaganych tematów z zakresu rachunku prawdopodobieństwa i statystyki matematycznej. Następnie opisujemy podstawowe metody uczenia maszynowego z probabilistycznego punktu widzenia. W końcu przechodzimy do zasadniczej części związanej z uczeniem głębokim i sieciami neuronowymi. W każdej części staramy się przedstawiać opisywane tematy w sposób minimalistyczny, skupiając się głównie na matematycznej i ideowej, a nie implementacyjnej stronie zagadnień. Liczymy, iż takie podejście zapewni odpowiednio głębokie zrozumienie tematu, dzięki któremu dalsze studiowanie całej gamy specyficznych technicznych tematów nie sprawi żadnego problemu.

1.1 Notacja

W dalszej części tekstu będziemy stosować przedstawioną tutaj pokrótce notację. Wektory, które traktujemy jako elementy przestrzeni $\mathbb{R}^d$ ze standardowo zdefiniowanymi operacjami dodawania i mnożenia przez skalar będziemy oznaczali wytłuszczonymi małymi lub wielkimi literami np. $\boldsymbol{x}, \boldsymbol{X}$. Wielkość $\boldsymbol{x}^i$ będzie oznaczać dany element wektora (w tym przypadku i–ty element $\boldsymbol{x}$). Wielkość $\boldsymbol{x}_\mu$ będzie oznaczać pewien (w tym przypadku μ–ty) element pewnego zbioru wektorów. Macierze oraz wielowymiarowe tablice (zwane również niefortunnie tensorami) będziemy oznaczać (jedynie) wytłuszczonymi wielkimi literami np. $\boldsymbol{X}, \boldsymbol{\Phi}$. Analogicznie jak w przypadku wektorów przez $\boldsymbol{X}^{i_1 i_2 \cdots i_k}$ będziemy oznaczać $(i_1, i_2, \ldots, i_k)$ element k–wymiarowej tablicy $\boldsymbol{X}$, natomiast $\boldsymbol{X}_\mu$ będzie oznaczać μ–ty element pewnego zbioru tablic.

1.2 Uczenie nadzorowane

Uczenie nadzorowane jest jednym z dwóch podstawowych (pomijając tzw. uczenie ze wzmocnieniem) paradygmatów w uczeniu maszynowym, którego ogólną ideą jest zdefiniowanie pewnego modelu odwzorowującego dane wejściowe na wyjściowe predykcje. Zakładamy w nim, iż mamy dostępny zbiór obserwacji $\mathcal{X} = \{y_i(\boldsymbol{x}_i)\}_{i=1}^n$, gdzie $\boldsymbol{x} \in \mathbb{R}^m$ nazywamy wektorem cech a $y(\boldsymbol{x})$ jest prawidłową wartością odpowiedzi dla tych cech. Dwa najbardziej podstawowe przypadki zagadnień tego rodzaju to regresja oraz klasyfikacja. W przypadku regresji zmienna y przyjmuje wartości z pewnego podzbioru liczb rzeczywistych. W przypadku klasyfikacji zmienna y przyjmuje wartości ze skończonego zbioru kategorii, przy czym wartości z tego zbioru nie powinny posiadać naturalnej tj. wynikającej z natury problemu, relacji porządku.

W jaki sposób tworzymy model odwzorowujący $\boldsymbol{x}$ na y? W dalszych paragrafach poznamy różne metody, ale najczęściej modelem jest pewna rodzina funkcji postaci $\phi(\boldsymbol{x}; \boldsymbol{w})$ parametryzowana skończoną liczbą parametrów, które możemy łącznie zapisać jako pewien wektor $\boldsymbol{w}$. Aby znaleźć parametry $\boldsymbol{w}$, dzięki którym dla konkretnego zagadnienia model będzie zadowalająco odwzorowywał cechy na predykcje (innymi słowy aby nauczyć model) wprowadzamy dodatkowo funkcjonał kosztu (z ang. *loss function*) $L(\mathcal{X}; \boldsymbol{w})$, który kwantyfikuje odpowiedzi modelu ϕ w stosunku do znanych prawidłowych odpowiedzi y dla danych ze zbioru $\mathcal{X}$. Najczęściej ma on postać

$$L(\mathcal{X}; \boldsymbol{w}) = -\frac{1}{n} \sum_{i=1}^{n} \log p\left(y_i(\boldsymbol{x}_i) \mid \boldsymbol{w}\right) ,$$

gdzie $p(y(\boldsymbol{x}) \mid \boldsymbol{w})$ jest warunkową gęstością prawdopodobieństwa danej obserwacji $y(\boldsymbol{x})$ warunkowaną przez wartość parametrów $\boldsymbol{w}$ (oczywiście gęstość ta zależy od estymowanych parametrów $\boldsymbol{w}$ przez funkcję $\phi(\boldsymbol{x}; \boldsymbol{w})$. Do powyższego funkcjonału możemy również dodawać tzw. człony regularyzujące (z ang. *regularizers*). Trening modelu polega wówczas na znalezieniu parametrów $\boldsymbol{w}^*$, które minimalizują funkcjonał kosztu na zbiorze treningowym $\mathcal{X}$

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} L(\mathcal{X}; \boldsymbol{w}) .$$

Zauważmy, że takie podejście ma jedną zasadniczą wadę – istotnie nie interesuje nas tak naprawdę, jak model radzi sobie na zbiorze treningowym (tzn. zagadnienie uczenia jest czymś więcej niż numeryczną minimalizacją funkcji), tylko jak będzie radził sobie na nowych, niewidzianych wcześniej danych (zależy nam przede wszystkim na generalizacji). Sytuację, w której model bardzo dobrze modeluje dane w zbiorze treningowym, ale słabo radzi sobie na nowych danych nazywamy przeuczeniem lub nadmiernym dopasowaniem (z ang. *overfitting*). Sytuację, w której model słabo radzi sobie zarówno na zbiorze treningowym, jak i na nowych danych nazywamy niedouczeniem lub niedopasowaniem (z ang. *underfitting*). Występowanie overfittingu i underfittingu jest powiązane z pojemnością (z ang. *capacity*) modelu. Złożony model o dużej pojemności potrafi dopasować się do bardzo skomplikowanych obserwacji (jest elastyczny), ale istnieje ryzyko jego przeuczenia (mówimy wówczas o *high variance*). Dla prostego modelu o małej pojemności istnieje z kolei ryzyko, iż nie ma on wystarczająco ekspresywności (mówimy wówczas o *high bias*).

1.3 Uczenie nienadzorowane

W przypadku uczenia nienadzorowanego naszym celem nie jest znalezienie modelu odwzorowującego cechy na predykcje. Chcemy raczej zrozumieć wewnętrzną strukturę danych oraz odkryć zależności między zmiennymi lub grupami zmiennych. Modele tego rodzaju znajdują zastosowanie w analizie biznesowej, gdzie pozwalają, chociażby na analizę ważności poszczególnych wskaźników, czy wizualizację wysoko-wymiarowych danych.

1.4 Praktyka uczenia maszynowego

W dalszej części skupiamy się przede wszystkim na matematycznej stronie prezentowanych zagadnień, ale należy pamiętać, iż dowolną próbę wdrożenia modelu uczenia maszynowego należy zacząć od dokładnej inspekcji danych, dla których przygotowujemy ów model („become one with the data”,

A. Karpathy), a zakończyć dogłębną analizą metryk pozwalających na ewaluację wytrenowanego modelu. Warunkiem koniecznym udanego wdrożenia modelu jest więc odpowiednie zebranie, analiza i przygotowanie danych, które trafiają następnie jako wejście do modelu ML, a następnie odpowiedni dobór i dogłębna analiza wyników ewaluacji modelu. W tym paragrafie pokrótce opisujemy elementarne praktyki, o których należy pamiętać przy wdrażaniu modeli ML.

1.4.1 Przygotowanie danych

Kluczem do uzyskania dobrych wyników przy korzystaniu z algorytmów uczenia maszynowego jest odpowiednie przygotowanie danych (z ang. *preprocessing*). Typowo preprocessing składa się z:

- eksploracji danych oraz wstępnego czyszczenia, w szczególności usunięcia jawnych wartości odstających (z ang. *outliers*) oraz cech posiadających zbyt dużo wartości brakujących;
- analizy rozkładu zmiennej docelowej oraz ewentualnej transformacji logarytmicznej, która poprawia stabilność numeryczną, gdy przewidywane wartości są dużymi dodatnimi liczbami rzeczywistymi, zmienia dziedzinę zmiennej objaśnianej z $\mathbb{R}_+$ na $\mathbb{R}$ oraz dodatkowo jest przykładem transformacji stabilizującej wariancję;
- podziału zbioru na część treningową oraz testową;
- dokonania skalowania i imputacji brakujących wartości cech (metody `.fit()` wywołujemy jedynie dla zbioru treningowego);
- usunięcia silnie skorelowanych cech;
- zakodowania wartości kategorycznych za pomocą tzw. *one–hot encoding* pamiętając o *dummy variable trap* – jedną z k kategorii kodujemy za pomocą wektora *one–hot* długości $n - 1$, aby uniknąć zależności liniowej między cechami (opcja `drop="first"` w `OneHotEncoder` w scikit-learn);
- wykonania feature engineering – dodania wielomianów cech do naszych danych lub skonstruowania innych cech (np. cech określających miesiąc, dzień itp.).

Podział zbioru na część treningową i testową jest najważniejszym etapem preprocessingu. Zbiór testowy wydzielamy, aby po wytrenowaniu modelu sprawdzić, jak poradzi on sobie na nowych, niewidzianych wcześniej

danych. Powinniśmy go traktować jako dane, które będziemy w przyszłości dostawać po wdrożeniu modelu do realnego systemu. Takie dane również będziemy musieli przeskalować, zakodować itp., ale parametry potrzebne do wykonania tych transformacji możemy wziąć jedynie z dostępnego wcześniej zbioru treningowego. Wykorzystanie danych testowych w procesie treningu to błąd wycieku danych (z ang. *data leakage*). Skutkuje on niepoprawnym, nadmiernie optymistycznym oszacowaniem jakości modelu.

1.4.2 Metryki do oceny regresji i klasyfikacji

Zasadniczo, aby ocenić predykcje modelu używamy odpowiednich metryk, których wartości określają jak dobry jest model.

W przypadku regresji najczęściej używanymi metrykami są RMSE (z ang. *Root Mean Squared Error*) oraz MAE (z ang. *Mean Absolute Error*) zdefiniowane odpowiednio jako

$$\mathrm{RMSE} := \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}\,, \quad \mathrm{MAE} := \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|\,.$$

Metryki te mają jednakową jednostkę jak predykcje. Jeśli chcielibyśmy mieć liczbę względną określającą jakość modelu to mamy do dyspozycji metryki MAPE (z ang. *Mean Absolute Percentage Error*) oraz SMAPE (z ang. *Symmetric Mean Absolute Percentage Error*) zdefiniowane odpowiednio jako

$$\mathrm{MAPE} := \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\,, \quad \mathrm{SMAPE} := \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}\,.$$

Obie te metryki mają zakres od 0 do 1, przy czym niższa wartość oznacza lepszy model. Metryki te mają jednak szereg problemów, z których najpoważniejsze to: problemy, gdy wartości są bliskie 0, asymetryczne traktowanie predykcji za dużych oraz za małych. Z tych powodów znacznie lepszą względną metryką jest MASE (z ang. *Mean Absolute Scaled Error*)

$$\mathrm{MASE} := \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\sum_{i=1}^{n}|y_i - \overline{y}|}\,,$$

gdzie $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Metryka MASE jest zatem względnym błędem MAE jaki popełnia nasz model w stosunku do modelu naiwnego, który przewiduje zawsze wartość średnią.

W przypadku zadania klasyfikacji binarnej naszym celem dla danego wektora cech jest zwrócenie jednej z dwóch klas, które będziemy nazywać

klasą pozytywną i negatywną. O ile w przypadku regresji pomiar jakości modelu był całkiem prosty, o tyle w przypadku klasyfikacji sytuacja jest nieco bardziej skomplikowana. Zauważmy bowiem, iż mamy 4 możliwości odpowiedzi klasyfikatora

- *True Positive (TP)* – poprawnie zaklasyfikowaliśmy klasę pozytywną jako pozytywną
- *True Negative (TN)* – poprawnie zaklasyfikowaliśmy klasę negatywną jako negatywną
- *False Positive (FP)* – niepoprawnie zaklasyfikowaliśmy klasę negatywną jako pozytywną
- *False Negative (FN)* – niepoprawnie zaklasyfikowaliśmy klasę pozytywną jako negatywną

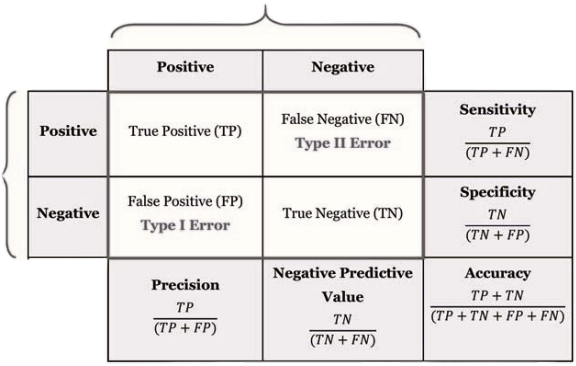
Na podstawie ilości TP, TN, FP i FN w zbiorze testowym możemy wykreślić tzw. macierz pomyłek (z ang. *confusion matrix*) pokazującą ilość każdej z możliwości. Następnie możemy obliczyć różne stosunki tych wartości, aby uzyskać różne metryki. Najbardziej standardowymi są accuracy, precision oraz recall (lub inaczej sensitivity) zdefiniowane jako

$$\text{Accuracy} := \frac{\text{TP} + \text{TN}}{n}, \quad \text{Precision} := \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} := \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Wartość accuracy mówi po prostu jaki stosunek przykładów został poprawnie zaklasyfikowany (zauważmy tutaj, że $\text{TP} + \text{TN} + \text{FP} + \text{FN} = n$). Nie jest to jednak dobra miara jakości, gdy nasz zbiór jest niezbalansowany, tj. zawiera więcej przykładów określonej klasy.

Wartość precision określa jak pewny jest klasyfikator przy wykrywaniu klasy pozytywnej, natomiast recall mówi o tym jak dobrze klasyfikator „wyławia" przykłady pozytywne. Zauważmy jednak, iż nie możemy stosować żadnej z tych metryk w odosobnieniu. Istotnie klasyfikator, który zwraca zawsze klasę pozytywną ma maksymalny recall, a klasyfikator, który zwraca zawsze klasę negatywną ma nieokreślony precision (i jest oczywiście beznadziejnym klasyfikatorem). Musimy więc zawsze ewaluować model na obu tych metrykach i jedynie dobry wynik obu z nich mówi o jakości klasyfikatora. Oczywiście czasami chcielibyśmy określić jakość modelu za pomocą jednej liczby, a niekoniecznie sprawdzać zawsze macierz pomyłek (choć jest to bardzo użyteczne) lub podawać wartości dwóch metryk. Metryką, która łączy precision i recall jest F_β–score zdefiniowany jako

$$F_\beta := (1 + \beta^2) \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}},$$



Rysunek 1: Macierz pomyłek oraz możliwe metryki oceny jakości klasyfikatora

gdzie β określa ile razy bardziej ważny jest recall od precision. Typowo używa się F_1–score

$$F_1 = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}\,.$$

Wiele klasyfikatorów oprócz twardych predykcji zwraca również rozkład prawdopodobieństwa nad klasami. W przypadku klasyfikacji binarnej jest to oczywiście rozkład zero-jedynkowy z parametrem p określającym prawdopodobieństwo klasy pozytywnej dla danego wektora cech. Standardowo oczywiście twardą predykcją jest ta z klas, która ma większe prawdopodobieństwo, czyli (co równoważne) predykcją jest klasa pozytywna jeśli $p > 0.5$. W niektórych problemach chcemy jednak zmienić ten próg i dokonać tzw. *threshold tuning*. Wykresem, który pozwala na dokonanie tuningu progu jest krzywa ROC (z ang. *Receiver Operatic Characteristic curve*), która jest krzywą parametryczną wyznaczoną przez punkty $(\text{FPR}(\text{threshold}), \text{TPR}(\text{threshold}))$ dla progów z zakresu $[0; 1]$, gdzie

$$\text{TPR} := \frac{\text{TP}}{\text{TP} + \text{FN}}\,, \quad \text{FPR} := \frac{\text{FP}}{\text{FP} + \text{TN}}\,.$$

Metryką niezależną od wybranego progu jest tzw. AUROC (z ang. *Area under ROC curve*) zdefiniowany jako pole powierzchni pod krzywą ROC dla danego klasyfikatora. Zauważmy, że klasyfikator losowy, który zwraca zawsze klasę pozytywną z prawdopodobieństwem równym wartości progu ma wartość AUROC równą 0.5, natomiast idealny klasyfikator, który nie-

zależnie od wartości progu klasyfikuje wszystkie przykłady poprawnie ma AUROC równy 1.

Inną analogiczną metryką jest AUPRC, gdzie zamiast krzywej ROC stosujemy krzywą PRC (z ang. *Precision–Recall Curve*), w której zamiast TPR i FPR używamy odpowiednio Precision i Recall. Metryka AUPRC jest często wykorzystywana w przypadku klasyfikacji ekstremalnie niezbalansowanej, w której mamy bardzo mało ($< 1\%$) klasy pozytywnej.

W przypadku klasyfikacji wieloklasowej używamy zasadniczo takich samych metryk jak w klasyfikacji binarnej, ale wprowadzamy mikro i makro uśrednianie (z ang. *micro/macro-averaging*). Przez TP_k będziemy rozumieć liczbę prawidłowo zaklasyfikowanych przykładów z klasy k, FP_k to liczba przykładów z innych klas, które zaklasyfikowaliśmy nieprawidłowo jako k–tą klasę, FN_k to liczba przykładów z klasy k, które zaklasyfikowaliśmy jako inną klasę. Wówczas odpowiednie metryki mają postać

$$\mathrm{MicroPrecision} := \frac{\sum_k \mathrm{TP}_k}{\sum_k \mathrm{TP}_k + \sum_k \mathrm{FP}_k},$$
$$\mathrm{MacroPrecision} := \frac{1}{K}\sum_{k=1}^{K} \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k}$$

oraz

$$\mathrm{MicroRecall} := \frac{\sum_k \mathrm{TP}_k}{\sum_k \mathrm{TP}_k + \sum_k \mathrm{FN}_k},$$
$$\mathrm{MacroRecall} := \frac{1}{K}\sum_{k=1}^{K} \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FN}_k}.$$

W przypadku klasyfikacji wieloklasowej macierz pomyłek jest macierzą wymiaru $K \times K$, gdzie K jest liczbą klas.

1.4.3 Tuning hiperparametrów i walidacja skrośna

Praktycznie wszystkie modele uczenia maszynowego mają hiperparametry, często liczne, które w zauważalny sposób wpływają na wyniki, a szczególnie na underfitting i overfitting. Ich wartości trzeba dobrać zatem dość dokładnie. Proces doboru hiperparametrów nazywa się tuningiem hiperparametrów (z ang. *hyperparameter tuning*).

Istnieje na to wiele sposobów. Większość z nich polega na tym, że trenuje się za każdym razem model z nowym zestawem hiperparametrów i wybiera się ten zestaw, który pozwala uzyskać najlepsze wyniki. Metody

głównie różnią się między sobą sposobem doboru kandydujących zestawów hiperparametrów. Najprostsze i najpopularniejsze to:

- pełne przeszukiwanie (z ang. *grid search*) – definiujemy możliwe wartości dla różnych hiperparametrów, a metoda sprawdza ich wszystkie możliwe kombinacje (czyli siatkę),
- losowe przeszukiwanie (z ang. *randomized search*) – definiujemy możliwe wartości jak w pełnym przeszukiwaniu, ale sprawdzamy tylko ograniczoną liczbę losowo wybranych kombinacji.

Jak ocenić, jak dobry jest jakiś zestaw hiperparametrów? Nie możemy sprawdzić tego na zbiorze treningowym – wyniki byłyby zbyt optymistyczne. Nie możemy wykorzystać zbioru testowego – mielibyśmy wyciek danych, bo wybieralibyśmy model explicite pod nasz zbiór testowy. Trzeba zatem osobnego zbioru, na którym będziemy na bieżąco sprawdzać jakość modeli dla różnych hiperparametrów. Jest to zbiór walidacyjny (z ang. *validation set*). Zbiór taki wycina się ze zbioru treningowego.

Jednorazowy podział zbioru na części nazywa się *split validation* lub *holdout*. Używamy go, gdy mamy sporo danych, i 10-20% zbioru jako dane walidacyjne czy testowe to dość dużo, żeby mieć przyzwoite oszacowanie. Zbyt mały zbiór walidacyjny czy testowy da nam mało wiarygodne wyniki – nie da się nawet powiedzieć, czy zbyt pesymistyczne, czy optymistyczne. W praktyce niestety często mamy mało danych. Trzeba zatem jakiejś magicznej metody, która stworzy nam więcej zbiorów walidacyjnych z tej samej ilości danych. Taką metodą jest walidacja skrośna (z ang. *cross validation*, CV). Polega na tym, że dzielimy zbiór treningowy na K równych podzbiorów, tzw. foldów. Każdy podzbiór po kolei staje się zbiorem walidacyjnym, a pozostałe łączymy w zbiór treningowy. Trenujemy zatem K modeli dla tego samego zestawu hiperparametrów i każdy testujemy na zbiorze walidacyjnym. Mamy K wyników dla zbiorów walidacyjnych, które możemy uśrednić (i ewentualnie obliczyć odchylenie standardowe). Takie wyniki są znacznie bardziej wiarygodne.

2 Probabilistyka

2.1 Zmienne losowe

Zmienna losowa to formalnie odwzorowanie ze zbioru zdarzeń elementarnych Ω tj. zbioru atomowych wyników doświadczenia losowego w zbiór $\mathbb{R}^n$

$$\boldsymbol{X} : \Omega \mapsto \mathbb{R}^n \,.$$

Jest to zatem funkcja, która przyporządkowuje zdarzeniom losowym wartość liczbową. Każda zmienna losowa opisuje więc zmienną w klasycznym sensie, której wartości pochodzi z pewnego rozkładu. Rozkład ten jest zadany jednoznacznie przez funkcję $F : \mathbb{R}^n \mapsto [0; 1]$ taką, że

$$F(\boldsymbol{x}) := \Pr\left(\boldsymbol{X}^1 \leq \boldsymbol{x}^1, \ldots, \boldsymbol{X}^n \leq \boldsymbol{x}^n\right),$$

którą nazywa się dystrybuantą (z ang. *cumulative distribution function, cdf*). Każdy rozkład zmiennej losowej można opisać za pomocą dystrybuanty jednak jest to niewygodne. W dwóch przypadkach: rozkładów dyskretnych i rozkładów ciągłych rozkład zmiennej losowej można opisać prościej za pomocą odpowiednio funkcji prawdopodobieństwa (z ang. *probability mass function, pmf*) oraz gęstości prawdopodobieństwa (z ang. *probability density function, pdf*).

Definicja 2.1. Zmienna losowa $\boldsymbol{X}$ ma dyskretny rozkład prawdopodobieństwa, jeśli istnieje skończony lub przeliczalny zbiór $\mathcal{S} \subset \mathbb{R}^n$ taki, że $\Pr(\boldsymbol{X} \in \mathcal{S}) = 1$. Wówczas rozkład ten jest zadany przez podanie funkcji prawdopodobieństwa $p(\boldsymbol{x}) = \Pr(\boldsymbol{X} = \boldsymbol{x})$ dla $\boldsymbol{x} \in \mathcal{S}$.

Definicja 2.2. Zmienna losowa $\boldsymbol{X}$ ma z kolei ciągły rozkład prawdopodobieństwa, jeśli istnieje funkcja $p : \mathbb{R}^n \mapsto \mathbb{R}_+$ taka, że

$$\Pr\left(\boldsymbol{X}^1 \in (a_1; b_1), \ldots, \boldsymbol{X}^n \in (a_n; b_n)\right) = \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} p(\boldsymbol{x}) \, \mathrm{d}^n \boldsymbol{x}$$

dla dowolnej kostki $(a_1; b_1) \times \ldots \times (a_n; b_n)$.

Zauważmy, że w przypadku rozkładu ciągłego zachodzi

$$F(\boldsymbol{x}) = \int_{-\infty}^{\boldsymbol{x}^1} \cdots \int_{-\infty}^{\boldsymbol{x}^n} p(\boldsymbol{x}') \, \mathrm{d}^n \boldsymbol{x}' \ .$$

Definicja 2.3 (Wartości oczekiwanej)**.** Wartością oczekiwaną funkcji zmiennej losowej $\boldsymbol{f}(\boldsymbol{X})$ nazywamy wektor $\mathbb{E}[\boldsymbol{f}(\boldsymbol{x})]$ określoną wzo-

rem

$$\sum_{\boldsymbol{x}\in\mathcal{S}} \boldsymbol{f}(\boldsymbol{x})p(\boldsymbol{x})\,, \quad \int\limits_{\mathbb{R}^n} \boldsymbol{f}(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}^n\boldsymbol{x}$$

odpowiednio dla rozkładu dyskretnego i ciągłego.

Definicja 2.4 (Macierzy kowariancji)**.** Macierzą kowariancji funkcji zmiennej losowej $\boldsymbol{f}(\boldsymbol{X})$ nazywamy macierz

$$\mathbb{E}\left[(\boldsymbol{f}(\boldsymbol{x})-\boldsymbol{m_f})(\boldsymbol{f}(\boldsymbol{x})-\boldsymbol{m_f})^T\right]\,,$$

gdzie

$$\boldsymbol{m_f} = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\,.$$

Elementy diagonalne macierzy kowariancji nazywamy wariancjami, a elementy pozadiagonalne kowariancjami.

Definicja 2.5 (Kwantyla i mody)**.** Kwantylem q_p rzędu $p \in (0;1)$ zmiennej losowej jednowymiarowej o rozkładzie ciągłym z dystrybuantą F nazywamy dowolne rozwiązanie równania

$$F(x) = p\,.$$

Modą tej zmiennej nazywamy dowolne maksimum lokalne gęstości tego rozkładu.

Twierdzenie 2.1. Niech zmienna n–wymiarowa $\boldsymbol{X}$ ma rozkład ciągły o gęstości $p_{\boldsymbol{X}}$ i niech $\boldsymbol{Y}^i = \boldsymbol{\varphi}^i(\boldsymbol{X})$ dla $i = 1,\ldots,n$. Jeśli odwzorowanie $\boldsymbol{\varphi}$ jest różniczkowalne i odwracalne, przy czym odwzorowanie odwrotne $\boldsymbol{\psi} = \boldsymbol{\varphi}^{-1}$ jest różniczkowalne, to n–wymiarowa zmienna $\boldsymbol{Y}$ ma rozkład o gęstości

$$p_{\boldsymbol{Y}}(\boldsymbol{y}) = |J| p_{\boldsymbol{X}}(\boldsymbol{\psi}(\boldsymbol{y}))\,,$$

gdzie $J := \det\left[\frac{\partial \boldsymbol{\psi}^j}{\partial \boldsymbol{y}^i}\right]$ jest jakobianem odwzorowania $\boldsymbol{\psi}$.

Twierdzenie 2.2. Dla dowolnych zmiennych losowych $\boldsymbol{X}$, $\boldsymbol{Y}$ zachodzi

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \int_{\mathbb{R}^k} p(\boldsymbol{x}, \boldsymbol{y}) \, \mathrm{d}^k \boldsymbol{y} \quad \text{(sum rule)}$$

Twierdzenie 2.3. Dla dowolnych zmiennych losowych $\boldsymbol{X}$, $\boldsymbol{Y}$ zachodzi

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x} \mid \boldsymbol{y}) p_{\boldsymbol{Y}}(\boldsymbol{y}) \quad \text{(product rule)}$$

Twierdzenie 2.4. Zmienne losowe $\boldsymbol{X}$ i $\boldsymbol{Y}$ są niezależne wtedy i tylko wtedy, gdy

$$p(\boldsymbol{x}, \boldsymbol{y}) = p_{\boldsymbol{X}}(\boldsymbol{x}) p_{\boldsymbol{Y}}(\boldsymbol{y}) \quad \text{(independence)}$$

Twierdzenie 2.5. Dla dowolnych zmiennych losowych $\boldsymbol{X}$, $\boldsymbol{Y}$ zachodzi

$$\underbrace{p(\boldsymbol{x} \mid \boldsymbol{y})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{y} \mid \boldsymbol{x})}^{\text{likelihood}} \overbrace{p_{\boldsymbol{X}}(\boldsymbol{x})}^{\text{prior}}}{\underbrace{\int_{\mathbb{R}^k} p(\boldsymbol{y} \mid \boldsymbol{x}) p_{\boldsymbol{X}}(\boldsymbol{x}) \, \mathrm{d}^k \boldsymbol{x}}_{\text{evidence}}} \quad \text{(Bayes theorem)}$$

Definicja 2.6 (Warunkowej wartości oczekiwanej)**.** Warunkową wartość oczekiwaną $\boldsymbol{f}(\boldsymbol{X})$ pod warunkiem $\boldsymbol{Y} = \boldsymbol{y}$ nazywamy wielkość

$$\mathbb{E}[\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{y}] = \int_{\mathbb{R}^n} \boldsymbol{f}(\boldsymbol{x}) p(\boldsymbol{x} \mid \boldsymbol{y}) \, \mathrm{d}^n \boldsymbol{x}$$

2.2 Ważne rozkłady jednowymiarowe

Definicja 2.7 (Rozkładu dwupunktowego)**.** Jeśli X jest zmienną losową rzeczywistą o rozkładzie dyskretnym i $\mathcal{S} = \{x_1, x_2\}$ oraz $p(x_1) = p$, to mówimy, że X ma rozkład dwupunktowy z parametrem p. Jeśli $x_1 = 1$ i $x_2 = 0$ to taki rozkład dwupunktowy nazywamy rozkładem zero–jedynkowym (lub rozkładem Bernoulliego) i oznaczamy jako $X \sim \mathrm{Ber}(p)$.

Definicja 2.8 (Schematu dwumianowego)**.** Rozważmy doświadczenie losowe o dwu możliwych wynikach: sukces osiągamy z prawdopodobieństwem p, porażkę z prawdopodobieństwem $1-p$. Doświadczenie tego rodzaju nazywamy próbą Bernoulliego. Doświadczenie takie jest modelowane zmienną losową o rozkładzie dwupunktowym z parametrem p. Schematem dwumianowym (lub schematem Bernoulliego) nazywamy doświadczenie polegające na n–krotnym powtórzeniu próby Bernoulliego, przy założeniu, iż poszczególne próby są od siebie niezależne.

Definicja 2.9 (Rozkładu dwumianowego)**.** Niech X będzie zmienną losową taką, że X jest liczbą sukcesów w schemacie dwumianowym długości n z prawdopodobieństwem sukcesu w każdej próbie równym p. Wówczas

$$\Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}\,.$$

Rozkład prawdopodobieństwa określony powyższym wzorem nazywam się rozkładem dwumianowym o parametrach n, p. Jeśli zmienna X ma rozkład dwumianowy to stosujemy notację $X \sim \mathrm{Bin}(n, p)$.

Definicja 2.10 (Rozkładu geometrycznego)**.** Mówimy, że zmienna losowa X ma rozkład geometryczny z parametrem $p \in (0;1)$, tj. $X \sim \mathrm{Geo}(p)$, jeśli $\mathcal{S} = \mathbb{N} \setminus \{0\}$, a funkcja prawdopodobieństwa ma postać

$$p(x) = (1-p)^{x-1} p\,.$$

Zmienna X opisuje czas oczekiwania na pierwszy sukces w schemacie

dwumianowym o nieskończonej długości.

Definicja 2.11 (Rozkładu Poissona)**.** Jeśli zmienna X o wartościach w $\mathbb{N}$ opisuje liczbę wystąpień pewnego powtarzalnego zdarzenia w przedziale czasowym $[0;t]$, przy czym spełnione są następujące założenia:

- powtórzenia zdarzenia występują niezależnie od siebie;
- „intensywność" wystąpień r jest stała;
- w danej chwili (rozumianej jako odpowiednio mały przedział) może zajść co najwyżej jedno zdarzenie

to zmienna ta ma rozkład Poissona z parametrem $\lambda = rt$, tj. $X \sim \mathrm{Pos}(\lambda)$. Jeśli $X \sim \mathrm{Pos}(\lambda)$, to

$$\Pr(X = k) = \frac{\mathrm{e}^{-\lambda}\lambda^k}{k!}\,.$$

Twierdzenie 2.6 (Poissona)**.** Niech (X_n) będzie ciągiem zmiennych losowych takich, że $X_n \sim \mathrm{Bin}(n, p_n)$, gdzie (p_n) jest ciągiem takim, że

$$\lim_{n\to\infty} np_n = \lambda$$

dla pewnej liczby $\lambda > 0$. Wówczas

$$\lim_{n\to\infty} \Pr(X_n = k) = \frac{\mathrm{e}^{-\lambda}\lambda^k}{k!}\,.$$

Definicja 2.12 (Rozkładu jednostajnego)**.** Mówimy, że zmienna X o rozkładzie ciągłym ma rozkład jednostajny na przedziale $[a;b]$ tzn. $X \sim \mathcal{U}(a,b)$ jeśli jej gęstość wyraża się wzorem

$$p(x) = \begin{cases} \frac{1}{b-a}\,, & x \in [a;b] \\ 0\,, & x \notin [a;b] \end{cases}\,.$$

Definicja 2.13 (Rozkładu wykładniczego)**.** Niech T będzie zmienną modelującą czas oczekiwania na pierwsze zdarzenie w ciągu zdarzeń takim, że czas wystąpienia każdego z nich w przedziale $[0;t]$ jest opisany przez zmienną $X \sim \mathrm{Pos}(\lambda t)$. wtedy

$$\Pr(T > t) = \Pr(X = 0) = \mathrm{e}^{-\lambda t}$$

oraz

$$\Pr(T > 0) = 1\,.$$

Mówimy wtedy, że T ma rozkład wykładniczy z parametrem λ, tzn. $T \sim \mathrm{Exp}(\lambda)$. Gęstość rozkładu wykładniczego ma postać

$$p(t) = \begin{cases} 0\,, & t \leq 0 \\ \lambda \mathrm{e}^{-\lambda t}\,, & t > 0 \end{cases}\,.$$

Definicja 2.14 (Rozkładu normalnego)**.** Mówimy, że zmienna losowa X o gęstości $p(x)$ ma rozkład normalny z parametrami $\mu \in \mathbb{R}, \sigma^2 \in [0;+\infty)$, tzn. $X \sim \mathcal{N}(\mu, \sigma^2)$, jeśli

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\,.$$

Jeśli $\mu = 0$ i $\sigma = 1$, to taki rozkład nazywamy standardowym rozkładem normalnym.

2.3 Wielowymiarowy rozkład normalny

Definicja 2.15 (Standardowego wielowymiarowego rozkładu normalnego)**.** Zmienna losowa $\boldsymbol{X}$ ma standardowy n–wymiarowy rozkład normalny jeśli jej składowe są niezależne i dla każdego $i = 1, \ldots, n$ $\boldsymbol{X}^i \sim \mathcal{N}(0,1)$. Jest to rozkład ciągły o gęstości

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}\right)\,.$$

Definicja 2.16 (Wielowymiarowego rozkładu normalnego)**.** Zmienna losowa $\boldsymbol{X}$ ma n–wymiarowy rozkład normalny (z ang. *Multivariate Normal Distribution, MVN*), tzn. $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ jeśli istnieje k–wymiarowa zmienna losowa $\boldsymbol{Z}$ o standardowym rozkładzie normalnym dla pewnego $k \leq n$ oraz istnieje $\boldsymbol{\mu} \in \mathbb{R}^n$ i macierz $\boldsymbol{A} \in \mathbb{R}^{n \times k}$ takie, że $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T$ oraz

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z} + \boldsymbol{\mu}\,.$$

Jeśli macierz $\boldsymbol{\Sigma}$ jest dodatnio określona, to rozkład $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ jest ciągły, a jego gęstość jest dana przez

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

Macierz $\boldsymbol{\Sigma}^{-1}$ nazywa się macierzą precyzji.

Poziomice gęstości niezdegenerowanego wielowymiarowego rozkładu normalnego są elipsoidami, których półosie są skierowane wzdłuż wektorów własnych macierzy $\boldsymbol{\Sigma}$ i mają długości proporcjonalne do pierwiastka z wartości własnych.

Twierdzenie 2.7 (Własności niezdegenerowanego rozkładu normalnego)**.** Niech $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ dla dodatnio określonej macierzy $\boldsymbol{\Sigma}$, wówczas

1. Wszystkie rozkłady brzegowe i warunkowe $\boldsymbol{X}$ są rozkładami normalnymi. Istotnie jeśli

   $$\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu_x}\\ \boldsymbol{\mu_y}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma_{xx}} & \boldsymbol{\Sigma_{xy}}\\ \boldsymbol{\Sigma_{yx}} & \boldsymbol{\Sigma_{yy}}\end{bmatrix}\right)$$

   to można pokazać, iż

   $$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu_x}, \boldsymbol{\Sigma_{xx}})\,, \quad \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu_y}, \boldsymbol{\Sigma_{yy}})\,.$$

   oraz $\boldsymbol{x} \mid \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu_{x|y}}, \boldsymbol{\Sigma_{x|y}})$, gdzie

   $$\boldsymbol{\mu_{x|y}} = \boldsymbol{\mu_x} + \boldsymbol{\Sigma_{xy}}\boldsymbol{\Sigma_{yy}^{-1}}(\boldsymbol{y}-\boldsymbol{\mu_y})\,, \quad \boldsymbol{\Sigma_{x|y}} = \boldsymbol{\Sigma_{xx}} - \boldsymbol{\Sigma_{xy}}\boldsymbol{\Sigma_{yy}^{-1}}\boldsymbol{\Sigma_{yx}}\,.$$

2. Zmienne składowe $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^n$ są niezależne wtedy i tylko wtedy, gdy $\boldsymbol{\Sigma}$ jest macierzą diagonalną.

2.4 Wnioskowanie statystyczne

Niech zmienna losowa $\boldsymbol{X}$ określa model rozkładu pewnej cechy (cech) w ustalonym zbiorze instancji (tzw. populacji generalnej). Innymi słowy, przyjmujemy, że wartości cech zachowują się jakby zostały wybrane losowo zgodnie z rozkładem zmiennej $\boldsymbol{X}$. Do podstawowych zagadnień wnioskowania statystycznego należą:

- oszacowanie wielkości charakteryzujących rozkład $\boldsymbol{X}$ (np. wartości średniej albo wariancji);
- weryfikacja hipotez dotyczących rozkładu $\boldsymbol{X}$ (tym nie będziemy się zajmować).

Definicja 2.17 (Modelu statystycznego)**.** Modelem statystycznym nazywamy parę $(\mathcal{P}, \mathcal{X})$, gdzie $\mathcal{P}$ jest rodziną rozkładów prawdopodobieństwa na zbiorze $\mathcal{X}$. Zazwyczaj przyjmuje się

$$\mathcal{P} = \{p(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$$

dla pewnego zbioru parametrów Θ. Model statystyczny nazywamy parametrycznym jeśli $\Theta \subset \mathbb{R}^k$.

Definicja 2.18 (Prostej próby losowej)**.** Prostą próbą losową o liczności n nazywamy ciąg niezależnych zmiennych losowych $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ o tym samym rozkładzie $p(\cdot \mid \boldsymbol{\theta}) \in \mathcal{P}$ (z ang. *independent and identically distributed, i.i.d*).

Definicja 2.19 (Estymatora)**.** Estymatorem nazywa się statystykę $\hat{\theta}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$ służącą do oszacowania wartości parametru θ. Liczbę $\hat{\theta}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ dla konkretnej realizacji prostej próby losowej nazywa się wartością estymatora albo estymatą.

Definicja 2.20 (Funkcji wiarygodności)**.** Funkcją wiarygodności (z ang. *likelihood function*) dla modelu $\mathcal{P} = \{p(\cdot \mid \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$ nazy-

wamy funkcję

$$\mathcal{L} : \mathbb{R}^n \times \Theta \ni (\boldsymbol{x}, \boldsymbol{\theta}) \mapsto \mathcal{L}(\boldsymbol{x}; \boldsymbol{\theta}) \in [0; +\infty)$$

wyznaczającą rozkład łączny obserwowanych danych jako funkcję parametru $\boldsymbol{\theta}$.

Niech $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ będzie prostą próbą losową. Jeśli $p(\cdot \mid \boldsymbol{\theta})$ opisuje rozkład warunkowy, z którego pochodzą obserwacje, to

$$\mathcal{L}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n; \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{x}_i \mid \boldsymbol{\theta}) \,.$$

Dla wygody obliczeń często rozważa się tzw. zanegowaną logarytmiczną funkcję wiarygodności (z ang. *Negated Log-Likelihood function, NLL*), tzn.

$$L(\boldsymbol{x}; \boldsymbol{\theta}) = -\log \mathcal{L}(\boldsymbol{x}; \boldsymbol{\theta}) \,.$$

Wówczas dla realizacji prostej próby losowej mamy

$$L(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n; \boldsymbol{\theta}) = -\sum_{i=1}^{n} \log p(\boldsymbol{x}_i \mid \boldsymbol{\theta}) \,.$$

Definicja 2.21 (Estymatora największej wiarygodności)**.** Estymatorem największej wiarygodności (z ang. *Maximum Likelihood Estimator, MLE*) nazywamy funkcję $\hat{\boldsymbol{\theta}}$, która przy ustalonych wartościach obserwacji (realizacji prostej próby losowej) $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ maksymalizuje wartość funkcji wiarygodności lub, co równoważne, minimalizuje wartość zanegowanej logarytmicznej funkcji wiarygodności tj.

$$\hat{\boldsymbol{\theta}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \arg\min_{\boldsymbol{\theta} \in \Theta} \left[-\sum_{i=1}^{n} \log p(\boldsymbol{x}_i \mid \boldsymbol{\theta}) \right] .$$

Jeśli funkcja wiarygodności jest różniczkowalna względem $\boldsymbol{\theta}$ dla dowolnych $\boldsymbol{x}^i$, to MLE można czasem wyznaczyć analitycznie korzystając z warunku koniecznego optymalności, tzn. rozwiązując układ równań

$$\frac{\partial L(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0 \,.$$

Jeśli MLE nie da się wyliczyć analitycznie, wyznacza się je przy użyciu algorytmów optymalizacji numerycznej. Estymatory MLE są asymptotycznie nieobciążone.

Definicja 2.22 (Estymatora MAP)**.** Estymatorem MAP (z ang. *Maximum a Posteriori Estimator, MAP*) nazywamy funkcję $\hat{\boldsymbol{\theta}}$, która przy ustalonych wartościach obserwacji (realizacji prostej próby losowej) $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ maksymalizuje wartość iloczynu funkcji wiarygodności i priora nad wartościami parametrów lub, co równoważne, minimalizuje zregularyzowaną logarytmiczną funkcję wiarygodności tj.

$$\hat{\boldsymbol{\theta}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \arg\min_{\boldsymbol{\theta} \in \Theta} \left[-\sum_{i=1}^{n} \log p(\boldsymbol{x}_i \mid \boldsymbol{\theta}) - n \log p(\boldsymbol{\theta}) \right] .$$

2.5 Liczby losowe w komputerze

Opiszemy jeszcze pokrótce metody generowania liczb pseudolosowych z dowolnych rozkładów prawdopodobieństwa w sposób algorytmiczny. Podstawowym narzędziem, którego będziemy potrzebować do generowania próbek z bardziej skomplikowanych rozkładów będzie prosty generator liczb z rozkładu jednostajnego $\mathcal{U}(0, 1)$. Moglibyśmy oczywiście wykorzystać jakieś fizyczne urządzenie lub proces, który generuje liczby prawdziwie losowe (np. detektor Geigera-Mullera, szum lamp elektronowych, ruletka), ale błędem byłaby rezygnacja z odtwarzalności. Poszukujemy zatem deterministycznej metody, która generuje sekwencje liczb, które są w przybliżeniu losowe. Podstawową metodą do algorytmicznego generowania liczb pseudolosowych jest tzw. liniowy generator kongruentny (z ang. *Linear Congruent Generator, LCG*), który jest opisany zależnością rekurencyjną

$$I_{j+1} = (aI_j + c) \mod m ,$$

gdzie a, c, m to pewne ustalone dodatnie liczby całkowite, a I_0 to tzw. ziarno (z ang. *seed*). LCG generuje liczby całkowite, więc w dużym uproszczeniu liczby zmiennoprzecinkowe z rozkładu $\mathcal{U}(0, 1)$ otrzymujemy jako I_j/m (trzeba tutaj jednak uwzględnić problemy wynikające z arytmetyki zmiennoprzecinkowej).

Mając już generator liczb z rozkładu jednostajnego $\mathcal{U}(0, 1)$ i znając jawny wzór na dystrybuantę $F(x)$ innego rozkładu jednowymiarowego $\mathcal{D}$ możemy generować liczby z tego rozkładu korzystając z tzw. metody odwrotnej

dystrybuanty. Istotnie, jeśli $U \sim \mathcal{U}(0,1)$, to $F^{-1}(U) \sim \mathcal{D}$. Istotnie

$$\Pr\big(F^{-1}(U) \le x\big) = \Pr(U \le F(x)) = F(x)\,.$$

Metoda ta ma jedną zasadniczą wadę – musimy znać jawny wzór na dystrybuantę $F(x)$. W przypadku np. tak ważnych rozkładów, jak rozkład normalny dystrybuanta nie jest funkcją elementarną i metoda ta nie jest najlepsza. W przypadku rozkładu normalnego znacznie lepszą metodą jest tzw. metoda Boxa–Mullera. Weźmy rozkład łączny dwóch niezależnych zmiennych losowych X, Y pochodzących ze standardowego rozkładu normalnego

$$p(x,y) = \frac{1}{2\pi}\exp\left(\frac{1}{2}(x^2+y^2)\right).$$

Skorzystamy ze wzoru na transformację zmiennych losowych. Istotnie niech

$$X = \sqrt{Z}\cos\Phi\,, \quad Y = \sqrt{Z}\sin\Phi\,,$$

dla $0 < Z$ oraz $0 \le \Phi < 2\pi$, wówczas

$$q(z,\phi) = p(x(z,\phi), y(z,\phi))\begin{vmatrix} \frac{1}{2\sqrt{z}}\cos\phi & \frac{1}{2\sqrt{z}}\sin\phi \\ -\sqrt{z}\sin\phi & \sqrt{z}\cos\phi \end{vmatrix} = \frac{1}{4\pi}\mathrm{e}^{\frac{z}{2}}\,.$$

Zauważmy, że otrzymaliśmy sferycznie symetryczny rozkład wykładniczy. Możemy zatem wylosować z rozkładu jednostajnego kąt ϕ oraz z rozkładu wykładniczego wartość z korzystając z metody odwrotnej dystrybuanty. Wówczas wartości $x = \sqrt{z}\cos\phi$, $y = \sqrt{z}\sin\phi$ będą pochodzić ze standardowego rozkładu normalnego. Aby wygenerować próbki z ogólnego wielowymiarowego rozkładu normalnego, korzystamy z definicji, tj. najpierw generujemy n próbek ze standardowego rozkładu normalnego, a następnie korzystamy z przekształcenia afinicznego $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{\mu}$, gdzie $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T$.

2.6 Monte Carlo

Umiemy już generować próbki z wielowymiarowego rozkładu normalnego. Chcemy teraz poznać metodę, która umożliwi generowanie próbek ze skomplikowanych, wielowymiarowych rozkładów prawdopodobieństwa, których gęstość znamy jedynie z dokładnością do stałej normalizującej, tj. znamy jedynie $\tilde{p}(\boldsymbol{x}) = Zp(\boldsymbol{x})$. Ograniczenie to wynika z chęci próbkowania z posteriora $p(\boldsymbol{x} \mid \boldsymbol{y})$ w sytuacji, gdy znamy jedynie rozkład łączny $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})$. Okazuje się, iż znajomość rozkładu jedynie z dokładnością do stałej normalizującej jest wystarczająca do generowania próbek z

tego rozkładu. Generowanie próbek z kolei wystarcza natomiast, na mocy silnego prawa wielkich liczb, do szacowania wartości średnich dowolnych funkcji zmiennej $\boldsymbol{x}$. Przypomnijmy, iż na mocy silnego prawa wielkich liczb ciąg średnich częściowych $(\overline{\boldsymbol{X}}_n)$ ciągu zmiennych losowych $(\boldsymbol{X}_n)$ i.i.d. z rozkładu $\boldsymbol{X} \sim \mathcal{D}$ jest zbieżny z prawdopodobieństwem 1 do wartości oczekiwanej $\mathbb{E}[\boldsymbol{X}]$ tj.

$$\Pr\left(\lim_{n\to\infty} \overline{\boldsymbol{X}}_n = \mathbb{E}[\boldsymbol{X}]\right) = 1\,.$$

Wartość oczekiwaną $\mathbb{E}[\boldsymbol{X}]$ możemy zatem przybliżyć średnią $\overline{\boldsymbol{X}}_n$ z dużej ilości próbek.

Pozostaje pytanie w jaki sposób generować próbki ze skomplikowanych rozkładów prawdopodobieństwa, których gęstości znamy jedynie z dokładnością do stałej normalizującej. Poniżej przedstawimy dwa algorytmy próbkowania: algorytm IS oraz Metropolisa–Hastingsa będący szczególną realizacją całej rodziny algorytmów próbkowania zwanych Markov Chain Monte Carlo (MCMC).

2.6.1 Algorytm Importance Sampling

Załóżmy, iż chcemy obliczyć wartość oczekiwaną pewnej funkcji zmiennej losowej $\boldsymbol{x}$ względem skomplikowanego rozkładu prawdopodobieństwa $p(\boldsymbol{x})$, który znamy jedynie z dokładnością do stałej normalizującej

$$p(\boldsymbol{x}) = \frac{1}{Z_p}\tilde{p}(\boldsymbol{x})$$

tj. szukamy

$$\mathbb{E}_p[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}^n\boldsymbol{x}\,.$$

Jeśli umiemy generować próbki $\boldsymbol{x}$ z innego (prostszego) rozkładu $q(\boldsymbol{x})$ (np. wielowymiarowego rozkładu normalnego), który nazywamy rozkładem proponującym kandydatów (z ang. *proposal distribution*) to możemy zapisać

$$\begin{aligned}\mathbb{E}_p[f(\boldsymbol{x})] &= \int\limits_{\mathbb{R}^n} f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}^n\boldsymbol{x} = \int\limits_{\mathbb{R}^n} f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\mathrm{d}^n\boldsymbol{x} \\ &= \mathbb{E}_q\left[f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] = \frac{Z_q}{Z_p}\mathbb{E}_q\left[f(\boldsymbol{x})\frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})}\right]\,.\end{aligned}$$

Zakładamy tutaj, iż nośnik rozkładu p zawiera się w nośniku q tj. $\operatorname{supp} p \subseteq \operatorname{supp} q$. Stosunek stałych Z_p/Z_q również możemy oszacować z próbek z q,

gdyż mamy

$$Z_p = \int\limits_{\mathbb{R}^n} \tilde{p}(\boldsymbol{x})\,\mathrm{d}^n\boldsymbol{x} = Z_q \int\limits_{\mathbb{R}^n} \frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})} q(\boldsymbol{x})\,\mathrm{d}^n\boldsymbol{x} = Z_q \mathbb{E}_q\left[\frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})}\right],$$

skąd ostatecznie

$$\mathbb{E}_p[f(\boldsymbol{x})] = \frac{\mathbb{E}_q\left[f(\boldsymbol{x})\frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})}\right]}{\mathbb{E}_q\left[\frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})}\right]}.$$

Jeśli z rozkładu q wygenerowaliśmy próbki $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ to na mocy silnego prawa wielkich liczb mamy

$$\boxed{\mathbb{E}_p[f(\boldsymbol{x})] \approx \frac{\sum_{i=1}^m f(\boldsymbol{x}_i)\frac{\tilde{p}(\boldsymbol{x}_i)}{\tilde{q}(\boldsymbol{x}_i)}}{\sum_{j=1}^m \frac{\tilde{p}(\boldsymbol{x}_j)}{\tilde{q}(\boldsymbol{x}_j)}} = \sum_{i=1}^m \lambda_i f(\boldsymbol{x}_i),}$$

gdzie

$$\boxed{\lambda_i = \frac{\tilde{p}(\boldsymbol{x}_i)/\tilde{q}(\boldsymbol{x}_i)}{\sum_{j=1}^m \tilde{p}(\boldsymbol{x}_j)/\tilde{q}(\boldsymbol{x}_j)}.}$$

Algorytm Importance Sampling jest prostym algorytmem Monte Carlo, który ma jeden zasadniczy problem. W jaki sposób mamy wybrać rozkład proponujący kandydatów q? Pewną odpowiedź na to pytanie sugeruje analiza wariancji statystyki

$$\overline{f}_m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) = \frac{1}{m}\sum_{i=1}^m \frac{f(\boldsymbol{x}_i)p(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i)}$$

dla $\boldsymbol{x}_i \sim q$ mamy

$$\mathbb{V}[\overline{f}_m] = \frac{1}{m}\mathbb{V}_q\left[f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] = \frac{1}{m}\int\limits_{\mathbb{R}^n} \frac{(f(\boldsymbol{x})p(\boldsymbol{x}) - \mu_f q(\boldsymbol{x}))^2}{q(\boldsymbol{x})}\,\mathrm{d}^n\boldsymbol{x}.$$

Chcemy oczywiście, aby wariancja była jak najmniejsza, gdyż wówczas mała liczba próbek da dobre przybliżenie wartości oczekiwanej. Rozkład proponujący kandydatów powinien być zatem proporcjonalny do $f(\boldsymbol{x})p(\boldsymbol{x})$, co może być trudne do praktycznego zrealizowania.

2.6.2 Algorytm Metropolisa–Hastingsa

Cała klasa algorytmów próbkowania MCMC opiera się na idei wyrażenia generowania próbek jako ewolucji pewnego łańcucha Markowa.

Definicja 2.23 (Łańcucha Markowa)**.** Łańcuchem Markowa nazwiemy ciąg zmiennych losowych $(\boldsymbol{X}_t)$ o wartościach w $\mathbb{R}^n$ taki, że spełnione jest kryterium Markowa

$$\begin{aligned}\forall A \subset \mathbb{R}^n : \Pr(\boldsymbol{X}_t \in A \mid \boldsymbol{X}_{t-1} = \boldsymbol{x}_{t-1}, \ldots, \boldsymbol{X}_0 = \boldsymbol{x}_0) \\ = \Pr(\boldsymbol{X}_t \in A \mid \boldsymbol{X}_{t-1} = \boldsymbol{x}_{t-1})\,.\end{aligned}$$

Elementy ciągu nazywamy stanami łańcucha.

Dany łańcuch jest zadany jednoznacznie przez podanie gęstości prawdopodobieństwa przejścia łańcucha ze stanu $\boldsymbol{x} \to \boldsymbol{y}$, którą będziemy oznaczać przez $\pi(\boldsymbol{y} \mid \boldsymbol{x})$ (zakładamy, iż prawdopodobieństwo przejścia jest niezależne od chwili t – łańcuch taki nazywamy jednorodnym). Funkcja π spełnia oczywiście warunek unormowania

$$\int_{\mathbb{R}^n} \pi(\boldsymbol{y} \mid \boldsymbol{x})\,\mathrm{d}^n\boldsymbol{y} = ,$$

istotnie prawdopodobieństwo przejścia gdziekolwiek ze stanu $\boldsymbol{x}$ jest równe 1. Będziemy zakładać dodatkowo, iż $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n : \pi(\boldsymbol{y} \mid \boldsymbol{x}) > 0$. Rozkład $p(\boldsymbol{x})$ łańcucha Markowa (tj. rozkład prawdopodobieństwa z którego losujemy stan łańcucha w danej chwili t) z daną funkcją przejścia π nazwiemy rozkładem stacjonarnym tego łańcucha jeśli

$$p(\boldsymbol{y}) = \int_{\mathbb{R}^n} \pi(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x})\,\mathrm{d}^n\boldsymbol{x}\ .$$

Rozkład stacjonarny danego łańcucha oznaczymy przez $p^*(\boldsymbol{x})$. Zauważmy, iż jeśli stan początkowy łańcucha $\boldsymbol{X}_0$ pochodzi z rozkładu stacjonarnego p^* to każdy kolejny stan $\boldsymbol{X}_t$ również pochodzi z rozkładu stacjonarnego. Jeśli z kolei stan początkowy pochodzi z jakiegoś innego rozkładu p_0 to rozkład łańcucha w chwili t jest dany przez relację rekurencyjną

$$p_t(\boldsymbol{y}) = \int_{\mathbb{R}^n} \pi(\boldsymbol{y} \mid \boldsymbol{x}) p_{t-1}(\boldsymbol{x})\,\mathrm{d}^n\boldsymbol{x}\ , \quad \text{dla } t > 1.$$

Rozkładem granicznym łańcucha Markowa nazwiemy granicę w sensie zbieżności punktowej

$$\lim_{t\to\infty} p_t(\boldsymbol{x})\,.$$

Przy podanych wyżej założeniach istnieje twierdzenie, które mówi iż taki łańcuch Markowa posiada jednoznaczny rozkład stacjonarny tożsamy z rozkładem granicznym. Ponadto warunkiem wystarczającym, aby dany rozkład $p(\boldsymbol{x})$ był rozkładem stacjonarnym łańcucha Markowa jest

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n : \pi(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x}) = \pi(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{y}) \,,$$

co wynika z scałkowania powyższego równania

$$\int\limits_{\mathbb{R}^n} \pi(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x}) \,\mathrm{d}^n \boldsymbol{x} = \int\limits_{\mathbb{R}^n} \pi(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{y}) \,\mathrm{d}^n \boldsymbol{x} = p(\boldsymbol{y}) \int\limits_{\mathbb{R}^n} \pi(\boldsymbol{x} \mid \boldsymbol{y}) \,\mathrm{d}^n \boldsymbol{x} = p(\boldsymbol{y}) \,.$$

Kryterium to nazywamy kryterium lokalnego balansu (z ang. *detailed balance condition*).

Podstawowa idea wykorzystania łańcuchów Markowa do generowania próbek ze skomplikowanego rozkładu p jest więc następująca: tworzymy łańcuch Markowa, dla którego p jest rozkładem stacjonarnym, wówczas rozpoczynając w dowolnym dopuszczalnym stanie początkowym $\boldsymbol{X}_0$ po wykonaniu dużej liczby kroków (etap ten nazywamy okresem przejściowym z ang. *burn-in period*) stan $\boldsymbol{X}_t$ (dla $t \gg 1$) tego łańcucha będzie w przybliżeniu pochodził z rozkładu granicznego p (nie jest jednak prosto stwierdzić po jak długim okresie przejściowym przybliżenie to jest wystarczająco dobre). Aby otrzymać z takiej procedury próbki prawdziwie i.i.d. każda z próbek musiałaby pochodzić z ponownego uruchomienia takiego łańcucha. Oczywiście jest to nieefektywne, więc w praktyce generujemy próbki z jednego łańcucha po prostu odrzucając pewne z nich tak aby uniknąć znaczących korelacji. Pozostaje pytanie jak skonstruować funkcję przejścia $\pi(\boldsymbol{y} \mid \boldsymbol{x})$ dla danego rozkładu granicznego $p(\boldsymbol{x})$. Podstawową konstrukcję podaje algorytm Metropolisa–Hastingsa.

Algorytm Metropolisa–Hastingsa

1. Jako stan początkowy przyjmij dowolną dopuszczalną wartość $\boldsymbol{x}_0$.
2. Będąc w aktualnym stanie $\boldsymbol{x}$ z prostego rozkładu proponującego kandydatów $q(\boldsymbol{y} \mid \boldsymbol{x})$ wylosuj kandydata $\boldsymbol{y}$ na wartość łańcucha w kolejnym stanie.
3. Z prawdopodobieństwem

$$r(\boldsymbol{y} \mid \boldsymbol{x}) = \min \left\{ 1, \frac{p(\boldsymbol{y}) q(\boldsymbol{x} \mid \boldsymbol{y})}{p(\boldsymbol{x}) q(\boldsymbol{y} \mid \boldsymbol{x})} \right\}$$

zaakceptuj kandydata jako nowy stan i przejdź do stanu $\boldsymbol{y}$. W przeciwnym razie pozostań w stanie $\boldsymbol{x}$.

4. GOTO 2.

Funkcja przejścia ma zatem postać

$$\pi_{\mathrm{MH}}(\boldsymbol{y} \mid \boldsymbol{x}) = q(\boldsymbol{y} \mid \boldsymbol{x}) r(\boldsymbol{y} \mid \boldsymbol{x})\,.$$

Pozostaje tylko wykazać, iż spełnione jest kryterium lokalnego balansu. Istotnie mamy

$$\begin{aligned} \pi_{\mathrm{MH}}(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x}) &= \min\{q(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x}), q(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{y})\} \\ \pi_{\mathrm{MH}}(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{y}) &= \min\{q(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{y}), q(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x})\} \end{aligned}\,,$$

skąd $\pi_{\mathrm{MH}}(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x}) = \pi_{\mathrm{MH}}(\boldsymbol{x} \mid \boldsymbol{y}) p(\boldsymbol{y})$. Zauważmy, iż nie musimy znać $p(\boldsymbol{x})$ z dokładnością do stałej normalizującej, gdyż

$$\frac{p(\boldsymbol{y})}{p(\boldsymbol{x})} = \frac{\tilde{p}(\boldsymbol{y})/Z_p}{\tilde{p}(\boldsymbol{x})/Z_p} = \frac{\tilde{p}(\boldsymbol{y})}{\tilde{p}(\boldsymbol{x})}\,.$$

Poza algorytmem Metropolisa–Hastingsa jest wiele innych algorytmów z rodziny MCMC. Większość z nich implementuje konkretny sposób generowania (zostawiając resztę struktury) tak, aby zmniejszyć korelację po okresie przejściowym i przyspieszyć zbieżność. Standardowo wykorzystywanymi algorytmami z tej klasy są algorytmy HMC (*Hamiltonian Monte Carlo*) oraz NUTS (*No U-Turn Sampler*).

2.7 Estymator jądrowy gęstości

W poprzednim paragrafie opisaliśmy w jaki sposób mając (nieznormalizowaną) funkcję gęstości prawdopodobieństwa $\tilde{p}(\boldsymbol{x})$ generować algorytmicznie próbki z opisanego przez nią rozkładu. Teraz zajmiemy się problemem odwrotnym tj. mając realizację prostej próby losowej z pewnego rozkładu chcemy znaleźć funkcję $\hat{p}(\boldsymbol{x})$, która estymuje gęstość rozkładu prawdopodobieństwa, z którego pochodzą próbki. Opiszemy tutaj jedną z najprostszych metod zwaną estymatorem jądrowym (z ang. *Kernel Density Estimator, KDE*). Estymatorem jądrowym gęstości funkcji p nazywamy funkcję

$$\hat{p}(\boldsymbol{x}) := \frac{1}{n} \sum_{i=1}^{n} K_{\boldsymbol{H}}(\boldsymbol{x} - \boldsymbol{x}_i)\,,$$

gdzie $\boldsymbol{H}$ jest symetryczną i dodatnio określoną macierzą zwaną *bandwidth matrix*, funkcja $K_{\boldsymbol{H}}$ ma postać

$$K_{\boldsymbol{H}}(\boldsymbol{x}) = (\det \boldsymbol{H})^{-1/2} K(\boldsymbol{H}^{-1/2}\boldsymbol{x}),$$

gdzie funkcja K zwana jądrem (z ang. *kernel*) jest gęstością prawdopodobieństwa pewnego sferycznie symetrycznego rozkładu wielowymiarowego. Wybór funkcji K nie jest kluczowy ze statystycznego punktu widzenia, więc możemy bez problemu założyć, iż jest to gęstość wielowymiarowego rozkładu normalnego, tj.

$$K_{\boldsymbol{H}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^m \det \boldsymbol{H}}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{H}^{-1}\boldsymbol{x}\right),$$

gdzie zakładamy $\boldsymbol{x} \in \mathbb{R}^m$. Większym problemem w przypadku KDE jest wybór odpowiedniego parametru $\boldsymbol{H}$. Jednym z prostszych wyborów w przypadku estymatora

$$\boxed{\hat{p}(\boldsymbol{x}) = \frac{1}{n\sqrt{(2\pi)^m \det \boldsymbol{H}}} \sum_{i=1}^{n} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_i)^T \boldsymbol{H}^{-1}(\boldsymbol{x}-\boldsymbol{x}_i)\right)}$$

jest tzw. reguła Silvermana, która podaje następujący przepis na macierz $\boldsymbol{H}$

$$\boldsymbol{H}^{ij} = 0, \quad \sqrt{\boldsymbol{H}^{ii}} = \left(\frac{4}{4+m}\right)^{\frac{1}{m+4}} n^{\frac{-1}{m+4}} \sigma_i,$$

gdzie σ_i jest estymatorem wariancji i-tej współrzędnej zmiennej $\boldsymbol{X}$. Inną możliwością jest tzw. reguła Scotta

$$\boldsymbol{H}^{ij} = 0, \quad \sqrt{\boldsymbol{H}^{ii}} = n^{\frac{-1}{m+4}} \sigma_i.$$

KDE w praktycznych zastosowaniach często przyspiesza się za pomocą odpowiednich struktur danych do wyszukiwania najbliższych sąsiadów w przestrzeni $\mathbb{R}^m$, tj. zamiast sumować przyczynki od wszystkich punktów $\boldsymbol{x}_i$ dla danego $\boldsymbol{x}$, znajdujemy jego k najbliższych sąsiadów $\boldsymbol{x}$ ze zbioru $\{\boldsymbol{x}_i\}_{i=1}^n$ stosując np. ANN i obliczamy przyczynki do $\hat{p}(\boldsymbol{x})$ tylko od nich.

3 Podstawy statystycznego uczenia maszynowego

Przechodzimy teraz do zagadnień uczenia maszynowego, w których wykorzystamy przedstawioną wcześniej teorię rachunku prawdopodobieństwa

(w szczególności teorię zmiennych losowych) oraz wnioskowania statystycznego.

3.1 Regresja liniowa

Rozpatrujemy teraz zagadnienie regresji tj. predykcji ciągłej wartości $y \in \mathbb{R}$ w zależności od wektora cech $\boldsymbol{x}$. Zakładamy ponadto prosty model liniowy, tj. nasze predykcje będą miały postać

$$\phi(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}$$

dla pewnych estymowanych parametrów $\boldsymbol{w}$. Zauważmy, iż taka postać modelu ϕ uwzględnia człon stały poprzez transformację $\boldsymbol{x} \leftarrow \begin{bmatrix} \boldsymbol{x}^T & 1 \end{bmatrix}^T$. W przypadku regresji liniowej parametry estymujemy za pomocą metody MLE dla następującego modelu statystycznego

$$y(\boldsymbol{x}) \mid \boldsymbol{w} \sim \mathcal{N}(\phi(\boldsymbol{x}; \boldsymbol{w}), \sigma^2) \,.$$

Niech $\mathcal{X} = \{y_i(\boldsymbol{x}_i)\}_{i=1}^n$ będzie naszym zbiorem obserwacji i.i.d. Wiarygodność ma zatem postać

$$\mathcal{L}(\mathcal{X}; \boldsymbol{w}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \prod_{i=1}^{n} \exp\left(-\frac{1}{2\sigma^2} \left[y_i - \phi(\boldsymbol{x}_i; \boldsymbol{w}) \right]^2 \right) .$$

Zanegowana logarytmiczna funkcja wiarygodności (funkcjonał kosztu) ma zatem postać

$$\boxed{L(\mathcal{X}; \boldsymbol{w}) = \frac{1}{2} \sum_{i=1}^{n} \left[y_i - \phi(\boldsymbol{x}_i; \boldsymbol{w}) \right]^2 \,,}$$

gdzie pominęliśmy multiplikatywne i addytywne człony stałe, gdyż nie wpływają one na zagadnienie optymalizacji. Minimalizując funkcję L względem $\boldsymbol{w}$ otrzymamy estymatę MLE tych parametrów. Otrzymana funkcja L ma postać formy kwadratowej, a otrzymany problem optymalizacyjny nazywamy metodą najmniejszych kwadratów (z ang. *Ordinary Least Squares, OLS*). Wprowadźmy macierz $\boldsymbol{X}^{ij} := \boldsymbol{x}_i^j$ oraz wektory $\boldsymbol{y}^i = y_i$, $\boldsymbol{\phi}^i = \phi(\boldsymbol{x}_i; \boldsymbol{w}) = \sum_j \boldsymbol{X}^{ij} \boldsymbol{w}^j$. Wówczas możemy zapisać funkcję kosztu jako

$$L(\mathcal{X}; \boldsymbol{w}) = \frac{1}{2} \sum_{i=1}^{n} \left(\boldsymbol{y}^i - \boldsymbol{\phi}^i(\boldsymbol{X}; \boldsymbol{w}) \right)^2 \,,$$

skąd

$$\frac{\partial L}{\partial \boldsymbol{w}^a} = \sum_b \frac{\partial L}{\partial \boldsymbol{\phi}^b} \frac{\partial \boldsymbol{\phi}^b}{\partial \boldsymbol{w}^a} \,.$$

Jednocześnie

$$\frac{\partial L}{\partial \boldsymbol{\phi}^b} = \boldsymbol{y}^b - \boldsymbol{\phi}^b\,, \quad \frac{\partial \boldsymbol{\phi}^b}{\partial \boldsymbol{w}^a} = \boldsymbol{X}^{ba}\,,$$

skąd

$$\frac{\partial L}{\partial \boldsymbol{w}^a} = \sum_b \boldsymbol{X}^{ba}(\boldsymbol{y}^b - \boldsymbol{\phi}^b)\,,$$

co możemy zapisać w kompaktowej formie macierzowej

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{\phi}) = \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\,.$$

Przyrównując powyższe równanie do 0 otrzymujemy następujący wzór na estymatę MLE parametrów $\boldsymbol{w}$

$$\boxed{\boldsymbol{w}_{\mathrm{MLE}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{X}^+\boldsymbol{y}\,,}$$

gdzie $\boldsymbol{X}^+$ oznacza pseudoodwrotność Moore'a–Penrose'a, którą można efektywnie obliczyć korzystając z rozkładu SVD macierzy $\boldsymbol{X}$.

3.2 Regularyzacja

Regularyzacją nazywamy proces polegający na wprowadzeniu ad hoc do zagadnienia optymalizacji dodatkowych członów tak, aby rozwiązanie było regularne (prostsze, nieosobliwe, jednoznaczne). W przypadku funkcji kosztu L najczęściej dodajemy człon penalizujący rozwiązania o dużej normie estymowanego parametru tj. człon postaci $\lambda\|\boldsymbol{w}\|$ dla pewnej normy $\|\cdot\|$ i hiperparametru λ zwanego siłą regularyzacji. W kontekście bayesowskim regularyzację można również rozumieć jako użycie estymacji MAP zamiast MLE dla odpowiednio dobranego priora nad estymowanymi parametrami.

Jeśli dla modelu regresji liniowej jako człon regularyzujący przyjmiemy $\lambda\|\boldsymbol{w}\|_2$, tj. normę L2 wektora estymowanych parametrów, to otrzymana funkcja kosztu ma postać

$$\boxed{L(\mathcal{X}; \boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n}\left[y_i - \phi(\boldsymbol{x}_i; \boldsymbol{w})\right]^2 + \lambda\|\boldsymbol{w}\|_2\,.}$$

Analogiczną postać można otrzymać rozważając estymatę MAP dla identycznego modelu statystycznego z priorem na wartości parametrów $\boldsymbol{w}$ będącym rozkładem normalnym $\mathcal{N}(0, \tau^2\mathbf{1})$ dla odpowiednio dobranego τ. Dla powyższej funkcji kosztu możemy bez problemu wyznaczyć wartość $\boldsymbol{w}$,

która ją minimalizuje. Istotnie korzystając z wyników dla modelu regresji liniowej mamy

$$\boldsymbol{w}_{\text{MAP}} = \left(\lambda \mathbf{1} + \boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y}\,,$$

Zagadnienie minimalizacji funkcji kosztu będącej formą kwadratową z dodanym członem regularyzującym w postaci normy L2 wektora nazywamy regresją grzbietową (z ang. *ridge regression*).

Innym przykładem regularyzacji jest tzw. regularyzacja L1, która polega na dodaniu do funkcji kosztu członu postaci $\lambda \sum_{j=1}^{d} |\boldsymbol{w}^j|$ tj. normy L1 wektora wag. Zagadnie optymalizacji formy kwadratowej z członem regularyzującym L1 nazywamy regresją LASSO. W takim przypadku nie da się prosto analitycznie znaleźć estymaty punktowej MAP i trzeba używać algorytmów optymalizacji numerycznej. W ogólności można połączyć regularyzacje L1 i L2 tj. rozważać zregularyzowaną funkcję kosztu postaci

$$\boxed{L(\mathcal{X}; \boldsymbol{w}) = \frac{1}{2} \sum_{i=1}^{n} \left[y_i - \hat{y}(\boldsymbol{x}_i; \boldsymbol{w})\right]^2 + \lambda_1 \|\boldsymbol{w}\|_1 + \lambda_2 \|\boldsymbol{w}\|_2^2\,.}$$

Zagadnienie minimalizacji takiej funkcji kosztu nazywamy ElasticNet i tak jak w przypadku LASSO musimy korzystać z algorytmów optymalizacji numerycznej. Często wykorzystuje się tutaj algorytmy bezgradientowe np. coordinate descent.

3.3 Regresja kwantylowa

Zastanówmy się wpierw na czym tak naprawdę polega modelowanie rozkładu warunkowego $y(\boldsymbol{x}) \mid \boldsymbol{w}$ za pomocą określonego rozkładu prawdopodobieństwa. Na pierwszy rzut oka może się wydawać, iż takie podejście wprowadza bardzo silne założenia, a co za tym idzie ograniczenia w stosowaniu naszego modelu. Zauważmy jednak, iż w przypadku podejścia typu *likelihood* rozkład jest niejako wybierany w taki sposób, aby jego parametry były użyteczne. Istotnie w przypadku regresji zwykle nie ma jak zweryfikować rzeczywistego rozkładu $y(\boldsymbol{x})$, gdyż mamy tylko po jednej wartości y dla danego $\boldsymbol{x}$. Modelując $y(\boldsymbol{x})$ rozkładem normalnym chodzi nam zatem raczej o to, że w takim modelu chcemy znaleźć parametr (prostą) taką, że masa prawdopodobieństwa punktów po obu stronach prostej jest jednakowa i większość masy jest zgromadzona w bliskiej odległości od prostej (wynika to z kształtu rozkładu normalnego, który nie posiada tzw. ciężkich ogonów)

W takim ujęciu możemy zakładać różne inne rozkłady na $y(\boldsymbol{x})$ jeśli interesują nas proste, które inaczej mają rozdzielać masę prawdopodobieństwa

między punkty lub też dopuszczać obserwacje leżące daleko od prostej regresji. Problemem w przypadku rozkładu normalnego jest jego czułość na wartości odstające, gdyż w rozkładzie normalnym ogony tego rozkładu mają stosunkowo niewielką masę prawdopodobieństwa. Chcielibyśmy zatem rozkład z ciężkimi ogonami (z ang. *heavy tails*). Dodatkowo chcielibyśmy mieć rozkład, który pozwala znaleźć prostą, która nie rozkłada masy po równo, ale np. tak, że 90% masy prawdopodobieństwa jest pod nią. Oba te problemy możemy rozwiązać modelując rozkład $y(\boldsymbol{x})$ przez tzw. asymetryczny rozkład Laplace'a (z ang. *Asymmetric Laplace Distribution, ALD*).

Definicja 3.1 (Asymetrycznego rozkładu Laplace'a)**.** Mówimy, iż zmienna losowa rzeczywista X ma asymetryczny rozkład Laplace'a, tzn. $X \sim \mathrm{ALD}(m, \lambda, q)$ jeśli jej gęstość wyraża się wzorem

$$p(x; m, \lambda, q) = \frac{q(1-q)}{\lambda} \begin{cases} \mathrm{e}^{-\frac{q-1}{\lambda}(x-m)}, & x \leq m \\ \mathrm{e}^{-\frac{q}{\lambda}(x-m)}, & x \geq m \end{cases} .$$

Zauważmy, że dystrybuanta rozkładu ALD ma postać

$$F(x; m, \lambda, q) = \begin{cases} q\mathrm{e}^{\frac{1-q}{\lambda}(x-m)}, & x \leq m \\ 1 - (1-q)\mathrm{e}^{-\frac{q}{\lambda}(x-m)}, & x \geq m \end{cases}$$

zatem parametr q określa rząd kwantyla m. W przypadku regresji możemy zatem modelować wartość $y(\boldsymbol{x})$ przez rozkład ALD dla ustalonego q postaci

$$y(\boldsymbol{x}) \mid \boldsymbol{w}, \lambda \sim \mathrm{ALD}(\phi(\boldsymbol{x}; \boldsymbol{w}), \lambda, q) ,$$

gdzie λ pełni podobną rolę jak σ w przypadku rozkładu normalnego. Widzimy wówczas, iż estymacja MLE $\boldsymbol{w}$ daje prostą regresji taką, że ułamek $1 - q$ masy prawdopodobieństwa znajduje się pod prostą (estymujemy zatem warunkowy kwantyl rzędu q). Zanegowana logarytmiczna funkcja wiarygodności (inaczej funkcja kosztu) dla modelu ALD ma postać

$$\boxed{L(\mathcal{X}; \boldsymbol{w}, \lambda) = \sum_{i=1}^{n} \left[(q-1) r_i \theta(-r_i) + q r_i \theta(r_i) \right] ,}$$

gdzie

$$r_i := y_i - \phi(\boldsymbol{x}_i; \boldsymbol{w}) ,$$

a θ oznacza funkcję skokową Heaviside'a. Taką funkcję kosztu nazywamy pinball loss, a otrzymane zagadnienie optymalizacji – regresją kwantylową.

Modelowanie $y(\boldsymbol{x})$ za pomocą rozkładu ALD pozwala nam w prosty i „robust" sposób znaleźć niepewność naszych predykcji punktowych, tj. dla danego zagadnienia dopasowujemy trzy modele oparte na ALD dla $q = 0.5$ (estymacja punktowa, mediana, odporna na outliery) oraz np. $q = 0.1$ i $q = 0.9$ będące oszacowaniem niepewności punktowej estymaty.

3.4 Procesy gaussowskie

Jak już wspomnieliśmy macierz kowariancji n–wymiarowej zmiennej losowej $\boldsymbol{x}$ o wartości oczekiwanej $\boldsymbol{\mu}$ jest zdefiniowana jako

$$\boldsymbol{\Sigma} = \mathbb{E}\left[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T\right] .$$

Wiemy również, iż macierz ta jest nieujemnie określona. Pokażemy teraz, iż dla każdej nieujemnie określonej macierzy symetrycznej $\boldsymbol{K}$ wymiaru $n \times n$ istnieje n–wymiarowa zmienna losowa o wielowymiarowym rozkładzie normalnym, dla której $\boldsymbol{K}$ jest macierzą kowariancji. Istotnie dla każdej nieujemnie określonej macierzy symetrycznej istnieje macierz $\boldsymbol{L}$ taka, że

$$\boldsymbol{K} = \boldsymbol{L}\boldsymbol{L}^T ,$$

jest to tzw. dekompozycja Choleskiego. Niech $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1})$, wówczas zmienna losowa $\boldsymbol{L}\boldsymbol{z}$ ma rozkład o zerowej wartości oczekiwanej i macierzy kowariancji

$$\mathbb{E}\left[(\boldsymbol{L}\boldsymbol{z})(\boldsymbol{L}\boldsymbol{z})^T\right] = \mathbb{E}\left[\boldsymbol{L}\boldsymbol{z}\boldsymbol{z}^T\boldsymbol{L}^T\right] = \boldsymbol{L}\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T]\boldsymbol{L}^T = \boldsymbol{L}\boldsymbol{1}\boldsymbol{L}^T = \boldsymbol{K} .$$

Powyższe własności wskazują, iż macierze kowariancji można w pewnym sensie utożsamiać z nieujemnie określonymi macierzami symetrycznymi.

Definicja 3.2 (Funkcji kowariancji)**.** Funkcję $k : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ taką, że $\forall m \in \mathbb{N} : \forall X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \subset \mathbb{R}^n$ macierz

$$k(X, X) = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_m) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2, \boldsymbol{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_m, \boldsymbol{x}_1) & k(\boldsymbol{x}_m, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_m, \boldsymbol{x}_m) \end{bmatrix}$$

jest dodatnio określoną macierzą symetryczną nazywamy funkcją kowariancji, jądrem dodatnio określonym (z ang. *positive definite kernel*) lub jądrem Mercera.

Dla dwóch zbiorów punktów $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \subset \mathbb{R}^n$ i $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s\} \subset \mathbb{R}^n$ i funkcji kowariancji k wprowadzimy oznaczenie

$$k(X,Y) := \begin{bmatrix} k(\boldsymbol{x}_1,\boldsymbol{y}_1) & k(\boldsymbol{x}_1,\boldsymbol{y}_2) & \cdots & k(\boldsymbol{x}_1,\boldsymbol{y}_s) \\ k(\boldsymbol{x}_2,\boldsymbol{y}_1) & k(\boldsymbol{x}_2,\boldsymbol{y}_2) & \cdots & k(\boldsymbol{x}_2,\boldsymbol{y}_s) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_m,\boldsymbol{y}_1) & k(\boldsymbol{x}_m,\boldsymbol{y}_2) & \cdots & k(\boldsymbol{x}_m,\boldsymbol{y}_s) \end{bmatrix}.$$

Poniżej podajemy kilka przykładów funkcji kowariancji

- *Gaussian kernel* dla normy $\|\cdot\|$ i hiper-parametrów a, l (amplituda i skala długości)

$$k(\boldsymbol{x},\boldsymbol{y}) = a^2 \exp\left\{-\frac{1}{2l^2}\|\boldsymbol{x}-\boldsymbol{y}\|^2\right\}$$

- *Periodic kernel* dla normy $\|\cdot\|$ i hiper-parametrów a, l, p (amplituda, skala długości, okres zmienności)

$$k(\boldsymbol{x},\boldsymbol{y}) = a^2 \exp\left\{-\frac{2}{l^2}\sin^2\left(\frac{\pi}{p}\|\boldsymbol{x}-\boldsymbol{y}\|\right)\right\}$$

- *White noise kernel* dla hiper-parametru σ

$$k(\boldsymbol{x},\boldsymbol{y}) = \sigma^2 \delta_{\boldsymbol{x},\boldsymbol{y}}$$

- *Matérn kernel* dla normy $\|\cdot\|$ i hiper-parametrów a, l, ν (amplituda, skala długości, regularność)

$$k(\boldsymbol{x},\boldsymbol{y}) = a^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}}{l}\|\boldsymbol{x}-\boldsymbol{y}\|\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}}{l}\|\boldsymbol{x}-\boldsymbol{y}\|\right),$$

gdzie $\Gamma(x)$ to funkcja gamma Eulera, a $K_\nu(x)$ to zmodyfikowana funkcja Bessela 2-go rodzaju rzędu ν.

Twierdzenie 3.1 (Własności funkcji kowariancji)**.** Suma lub iloczyn dwóch funkcji kowariancji oraz złożenie funkcji kowariancji z wielomianem o nieujemnych współczynnikach jest również funkcją kowariancji.

Definicja 3.3 (Procesu gaussowskiego)**.** Procesem Gaussowskim (z ang. *Gaussian Process*) nazywamy rodzinę skalarnych zmiennych losowych indeksowanych przez punkty $\boldsymbol{x} \in \mathbb{R}^n$

$$\mathcal{GP} = \{f_{\boldsymbol{x}} \mid \boldsymbol{x} \in \mathbb{R}^n\}$$

taką że każdy skończony podzbiór $\mathcal{GP}$ ma łącznie wielowymiarowy rozkład normalny tj. dla dowolnego zbioru $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \subset \mathbb{R}^n$ zachodzi

$$\begin{bmatrix} f_{\boldsymbol{x}_1} \\ \vdots \\ f_{\boldsymbol{x}_m} \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}_X, \boldsymbol{\Sigma}_X)\,.$$

Zauważmy, iż proces Gaussowski możemy jednoznacznie zdefiniować podając przepisy na parametry $\boldsymbol{\mu}_X$ i $\boldsymbol{\Sigma}_X$ dla dowolnego zbioru X. W praktyce często przyjmujemy $\boldsymbol{\mu}_X = \mathbf{0}$, natomiast przepisem na macierz kowariancji może być zdefiniowana wyżej funkcja kowariancji $k(X, X)$ tj.

$$\begin{bmatrix} f_{\boldsymbol{x}_1} \\ \vdots \\ f_{\boldsymbol{x}_m} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, k(X, X))\,.$$

Process Gaussowski daje nam w praktyce rozkład prawdopodobieństwa nad funkcjami $f : \mathbb{R}^n \mapsto \mathbb{R}$, których charakter jest określony przez jądro k (np. funkcja gładka dla jądra Gaussowskiego, okresowa dla jądra periodycznego, itp.). Zauważmy, że nie wnioskujemy tu o parametrach konkretnej rodziny funkcji (jak w przypadku regresji liniowej); interesuje nas jedynie rozkład predykcyjny. Załóżmy, iż w dokładnie znanych przez nas punktach $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ zaobserwowaliśmy wartości pewnej funkcji, o których zakładamy, iż pochodzą z procesu Gaussowskiego zadanego jądrem k, które wyraża nasze założenia a priori co do charakteru badanej funkcji

$$\boldsymbol{f}_X = \begin{bmatrix} f_{\boldsymbol{x}_1} \\ \vdots \\ f_{\boldsymbol{x}_m} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, k(X, X))\,.$$

Powiedzmy, iż chcemy znać wartości $\boldsymbol{f}_Y$ tej funkcji w zadanych punktach $Y = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_s\}$. Ponieważ założyliśmy, iż wartości funkcji pochodzą z procesu Gaussowskiego, więc rozkład łączny $\boldsymbol{f}_X$ i $\boldsymbol{f}_Y$ jest rozkładem nor-

malnym

$$\begin{bmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_Y \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(X,X) & k(X,Y) \\ k(Y,X) & k(Y,Y) \end{bmatrix}\right) .$$

Zauważmy, iż z twierdzenia o własnościach niezdegenerowanego rozkładu normalnego wnioskujemy, iż warunkowy $\boldsymbol{f}_Y \mid \boldsymbol{f}_X$ jest również rozkładem normalnym o parametrach

$$\begin{aligned} \boldsymbol{\mu} &= k(Y,X)k^{-1}(X,X)\boldsymbol{f}_X \\ \boldsymbol{\Sigma} &= k(Y,Y) - k(Y,X)k^{-1}(X,X)k(X,Y) \end{aligned} .$$

Dodatkową niepewność związaną z pomiarem wartości $\boldsymbol{f}_X$ możemy uchwycić zmieniając postać jądra

$$k(\boldsymbol{x},\boldsymbol{y}) \leftarrow k(\boldsymbol{x},\boldsymbol{y}) + \mathcal{I}_X(\boldsymbol{x})\sigma^2\delta_{\boldsymbol{x},\boldsymbol{y}} ,$$

gdzie σ jest hiper-parametrem określającym precyzję pomiaru. Oczywiście k jest dalej funkcją kowariancji, gdyż takie podstawienie powoduje jedynie dodanie dodatnich członów do pewnych elementów diagonalnych macierzy kowariancji, więc macierz ta jest nadal symetryczna i dodatnio określona. Wówczas rozkład predykcyjny ma parametry

$$\boxed{\begin{aligned} \boldsymbol{\mu} &= k(Y,X)\left[k(X,X) + \sigma^2\mathbf{1}\right]^{-1}\boldsymbol{f}_X \\ \boldsymbol{\Sigma} &= k(Y,Y) - k(Y,X)\left[k(X,X) + \sigma^2\mathbf{1}\right]^{-1}k(X,Y) \end{aligned}} .$$

3.5 Klasyfikator najbliższych sąsiadów

Do tej pory zajmowaliśmy się problemami regresji ciągłej zmiennej skalarnej. Przechodzimy teraz do metod uczenia maszynowego wykorzystywanych dla problemów klasyfikacji. Jako pierwszy rozważmy jeden z najprostszych (ale bardzo mocnych) modeli – klasyfikator k najbliższych sąsiadów (z ang. *k Nearest Neighbors, kNN*). Załóżmy, iż mamy zbiór obserwacji i.i.d. postaci $\mathcal{X} = \{y_i(\boldsymbol{x}_i)\}_{i=1}^n$ przy czym y może być zarówno skalarną zmienną ciągłą jak i elementem skończonego zbioru klas. Zakładamy, że wektory cech $\boldsymbol{x} \in \mathbb{R}^d$, a $\mathsf{d} : \mathbb{R}^d \times \mathbb{R}^d \mapsto I \subseteq \mathbb{R}$ jest metryką, półmetryką lub pewną miarą podobieństwa między punktami $\mathbb{R}^d$. Reguła decyzyjna klasyfikatora k najbliższych sąsiadów polega na znalezieniu dla nowego wektora cech $\boldsymbol{t}$, k najbliższych względem funkcji d punktów ze zbioru $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ i zwróceniu najczęstszej klasy dla tych sąsiadów. Metodę najbliższych sąsiadów można również wykorzystać do regresji, gdzie zwracamy wartość średnią

arytmetyczną wartości y dla znalezionych k najbliższych sąsiadów (powoduje to, że model taki potrafi tylko interpolować wartości, więc nie jest dobrym modelem dla regresji). W przypadku klasyfikacji możemy natomiast zwracać również rozkład prawdopodobieństwa klas dla nowego wektora cech $\boldsymbol{t}$ przez podanie stosunków występowania danej klasy wśród k najbliższych sąsiadów do k.

Klasyfikator kNN jest bardzo elastycznym modelem z nieliniową granicą decyzyjną. Jakość klasyfikacji silnie zależy od lokalnej gęstości punktów w przestrzeni $\mathbb{R}^d$ oraz wybranej wartości k, będącej hiperparametrem tego modelu. Ogólnie niskie k powoduje, że kNN ma duży variance i dość „poszarpaną" granicę decyzyjną, natomiast wysokie k powoduje, że kNN ma duży bias i „gładką" granicę decyzyjną. Typowo wykorzystywane funkcje d to

- metryka euklidesowa $\mathsf{d}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y})}$;
- pół-metryka euklidesowa $\mathsf{d}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y})$;
- metryka Manhattan $\mathsf{d}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} |\boldsymbol{x}^i - \boldsymbol{y}^i|$;
- podobieństwo cosinusowe $\mathsf{d}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^T \boldsymbol{y}}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2}$

Jedną z modyfikacji klasyfikatora kNN, który może polepszyć wyniki w przypadku, gdy w naszej przestrzeni istnieją obszary, w których mamy małą gęstość punktów ze zbioru $\mathcal{X}$ jest ważenie sąsiadów, które polega na tym, iż prawdopodobieństwa danej klasy wśród k sąsiadów obliczamy teraz jako średnią ważoną, gdzie wagą jest odwrotność odległości danego sąsiada od nowego wektora cech $w(\boldsymbol{t}, \boldsymbol{x}) = 1/\mathsf{d}(\boldsymbol{t}, \boldsymbol{x})$.

W przypadku naiwnego kNN podczas treningu zapamiętujemy jedynie zbiór $\mathcal{X}$ natomiast naiwna implementacja predykcji ma złożoność czasową $O(knd)$, przy założeniu, że złożoność obliczenia funkcji d dla pary punktów ma złożoność $O(d)$. Jest to nieakceptowalna złożoność, gdyż zwykle chcemy używać klasyfikatora kNN do setek milionów punktów. Dwa podejścia, które stosuje się zwykle do rozwiązania tego problemu to:

- zbudowanie odpowiedniej struktury danych w fazie treningu, aby w czasie predykcji można było szybciej znajdywać najbliższych sąsiadów (np. k-d tree, ball tree);
- wykorzystanie algorytmów aproksymacyjnych (z ang. *Approximate Nearest Neighbors, ANN*) do znajdowania sąsiadów, którzy niekoniecznie naprawdę są najbliżsi, ale aproksymacja jest wystarczająco dobra do praktycznych zastosowań.

Obecnie to drugie podejście jest dominujące i wykorzystywane na przykład w przypadku dużych modeli językowych do wyszukiwania kontekstów dla danych zapytań (z ang. *Retrieval Augmented Generation*).

3.6 Naiwny klasyfikator bayesowski

Rozważamy dalej problem klasyfikacji, w którym mamy zbiór przykładów i.i.d. postaci $\mathcal{X} = \{y_i(\boldsymbol{x}_i)\}_{i=1}^n$, gdzie $y \in \{c_1, \ldots, c_K\}$ oraz $\boldsymbol{x} \in \mathbb{R}^d$. W ogólności mamy rozkład warunkowy danej klasy pod warunkiem wektora cech, który jest dany przez gęstość prawdopodobieństwa $p(c_k \mid \boldsymbol{x})$. Korzystając z twierdzenia Bayesa możemy zapisać

$$p(c_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid c_k)p(c_k)}{p(\boldsymbol{x})}\,. \tag{3.6.1}$$

Jeśli umiemy obliczyć licznik wyrażenia po prawej stronie, to korzystając z reguły decyzyjnej MAP klasę dla nowego wektora cech wybieramy jako

$$\arg \max_{c_k \in \{c_1, \ldots, c_K\}} p(\boldsymbol{x} \mid c_k)p(c_k)\,. \tag{3.6.2}$$

Powstaje pytanie jak obliczyć to wyrażenie przy tak luźnych założeniach. Wprowadzamy naiwne założenie warunkowej niezależności cech względem danej klasy tj.

$$p(\boldsymbol{x} \mid c_k) = p(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^d \mid c_k) = \prod_{j=1}^{d} p(\boldsymbol{x}^j \mid c_k)\,. \tag{3.6.3}$$

Teraz jednowymiarowe rozkłady warunkowe $p(\boldsymbol{x}^j \mid c_k)$ możemy estymować z danych $\mathcal{X}$ np. korzystając z jądrowego estymatora gęstości lub zakładając konkretny model parametryczny (np. jednowymiarowy rozkład normalny, rozkład dwumianowy) i estymując jego parametry dla każdej z klas osobno. Naiwne założenie warunkowej niezależności pozwala złagodzić problemy wynikające z przekleństwa wymiarowości (z ang. *curse of dimensionality*), takie jak potrzeba zbiorów danych skalujących się wykładniczo wraz z liczbą cech d. Człon $p(c_k)$ można natomiast prosto oszacować jako stosunek przykładów danej klasy w zbiorze $\mathcal{X}$.

3.7 Wieloklasowa regresja logistyczna

Powróćmy do zagadnienia klasyfikacji, w którym mamy zbiór obserwacji i.i.d. $\mathcal{X} = \{y_i(\boldsymbol{x}_i)\}_{i=1}^n$, gdzie $y \in \{c_1, \ldots, c_K\}$ i $\boldsymbol{x} \in \mathbb{R}^d$. Pokażemy teraz podejście oparte na estymacji MLE podobnie jak w przypadku regresji.

Załóżymy, że obserwacje $y(\boldsymbol{x})$ pochodzą z rozkładu kategorialnego (wielopunktowego) zależnego od parametrów $\boldsymbol{W}$ jako

$$y(\boldsymbol{x}) \mid \boldsymbol{W} \sim \mathrm{Cat}\left(\boldsymbol{\pi}^1(\boldsymbol{\phi}(\boldsymbol{x};\boldsymbol{W})), \ldots, \boldsymbol{\pi}^K(\boldsymbol{\phi}(\boldsymbol{x};\boldsymbol{W}))\right) , \tag{3.7.1}$$

gdzie funkcja $\boldsymbol{\pi} : \mathbb{R}^K \mapsto [0;1]^K$ musi spełniać warunek unormowania

$$\forall \boldsymbol{z} \in \mathbb{R}^K : \sum_{j=1}^{K} \boldsymbol{\pi}^j(\boldsymbol{z}) = 1 ,$$

aby jej składowe były odpowiadały prawdopodobieństwom danej klasy, Funkcja $\boldsymbol{\phi} : \mathbb{R}^d \mapsto \mathbb{R}^K$ jest natomiast dowolną funkcją przekształcającą wektor cech na rzeczywisty wektor wymiaru K. W szczególności przyjmiemy następującą postać funkcji $\boldsymbol{\pi}$ zwaną funkcją softmax

$$\boxed{\boldsymbol{\pi}^j(\boldsymbol{z}) = \frac{\exp\left(\boldsymbol{z}^j\right)}{\sum_{k=1}^{K} \exp\left(\boldsymbol{z}^k\right)} .} \tag{3.7.2}$$

W przypadku wieloklasowej regresji logistycznej (zwanej również regresją softmax) jako funkcję $\boldsymbol{\phi}$ przyjmiemy proste przekształcenie liniowe

$$\boldsymbol{\phi}(\boldsymbol{x};\boldsymbol{W}) = \boldsymbol{W}\boldsymbol{x} , \tag{3.7.3}$$

gdzie $\boldsymbol{W}$ jest macierzą estymowanych parametrów wymiaru $K \times d$. Wyprowadzimy teraz wzór na funkcję kosztu przy takim modelu statystycznym. Zauważmy wpierw, iż wiarygodność pojedynczego przykładu ze zbioru $\mathcal{X}$ możemy zapisać jako

$$p(y_i(\boldsymbol{x}_i) \mid \boldsymbol{W}) = \prod_{j=1}^{K} \boldsymbol{\pi}^j \left(\boldsymbol{\phi}(\boldsymbol{x}_i;\boldsymbol{W})\right)^{[y_i=c_j]} , \tag{3.7.4}$$

gdzie $[\ldots]$ oznacza nawias Iversona. W takim razie wiarygodność całego zbioru $\mathcal{X}$ ma postać

$$\mathcal{L}(\mathcal{X};\boldsymbol{W}) = \prod_{i=1}^{n} \prod_{j=1}^{K} \boldsymbol{\pi}^j \left(\boldsymbol{\phi}(\boldsymbol{x}_i;\boldsymbol{W})\right)^{[y_i=c_j]} , \tag{3.7.5}$$

skąd funkcja kosztu ma postać

$$L(\mathcal{X};\boldsymbol{W}) = -\sum_{i=1}^{n} \sum_{j=1}^{K} [y_i = c_j] \log\left[\boldsymbol{\pi}^j \left(\boldsymbol{\phi}(\boldsymbol{x}_i;\boldsymbol{W})\right)\right] . \tag{3.7.6}$$

Wprowadzając macierze

$$\boldsymbol{X}^{ij} = \boldsymbol{x}_j^i\,, \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_{\boldsymbol{n}} \end{bmatrix}$$
$$\boldsymbol{T}^{ij} = [y_j = c_i]$$
$$\boldsymbol{\Phi}^{ij}(\boldsymbol{X};\boldsymbol{W}) = \boldsymbol{\phi}^i(\boldsymbol{x}_j;\boldsymbol{W}) = \sum_{k=1}^{d} \boldsymbol{W}^{ik}\boldsymbol{X}^{kj}\,, \quad \boldsymbol{\Phi}(\boldsymbol{X};\boldsymbol{W}) = \boldsymbol{W}\boldsymbol{X}$$
$$\boldsymbol{\Pi}^{ij}(\boldsymbol{\Phi}) = \boldsymbol{\pi}^i\left(\boldsymbol{\phi}(\boldsymbol{x}_j;\boldsymbol{W})\right) = \frac{\exp \boldsymbol{\Phi}^{ij}}{\sum_{k=1}^{K} \exp \boldsymbol{\Phi}^{kj}}$$

gdzie każda kolumna macierzy $\boldsymbol{X}$ jest wektorem cech danego przykładu; każda kolumna $\boldsymbol{T}$ jest tzw. wektorem one–hot dla danego przykładu tj. wektorem binarnym, w którym dokładnie na jednej pozycji jest wartość 1 i pozycja ta odpowiada prawidłowej klasie dla danego przykładu; każda kolumna macierzy $\boldsymbol{\Phi}$ jest wektorem mlogitów tj. liczb rzeczywistych, które po zastosowaniu funkcji softmax dają wartości prawdopodobieństwa każdej klasy; możemy zapisać

$$\boxed{L(\mathcal{X};\boldsymbol{W}) = -\sum_{j=1}^{n}\sum_{i=1}^{K} \boldsymbol{T}^{ij} \log\left[\boldsymbol{\Pi}^{ij}(\boldsymbol{\Phi}(\boldsymbol{X};\boldsymbol{W}))\right]\,.} \tag{3.7.7}$$

Niestety dla tak zdefiniowanej funkcji kosztu nie można znaleźć wzoru na minimum w postaci analitycznej (jak w przypadku regresji liniowej), dlatego do znalezienia estymaty MLE wykorzystujemy algorytmy optymalizacji numerycznej, w tym przypadku zwykle algorytmy gradientowe. Wyprowadzimy więc jeszcze wzór na pochodną funkcji kosztu po parametrach $\boldsymbol{W}$. Obliczmy najpierw pochodną L po $\boldsymbol{\Phi}^{ij}$ (dla przejrzystości zapisu nie piszemy granic sumowania – wynikają one naturalnie z wymiarów macierzy)

$$\frac{\partial L}{\partial \boldsymbol{\Phi}^{pq}} = -\sum_{i,j,r,s} \frac{\boldsymbol{T}^{ij}}{\boldsymbol{\Pi}^{rs}} \delta_{ir}\delta_{js} \frac{\partial \boldsymbol{\Pi}^{rs}}{\partial \boldsymbol{\Phi}^{pq}} = -\sum_{i,j} \frac{\boldsymbol{T}^{ij}}{\boldsymbol{\Pi}^{ij}} \frac{\partial \boldsymbol{\Pi}^{ij}}{\partial \boldsymbol{\Phi}^{pq}}\,. \tag{3.7.8}$$

Jednocześnie

$$\frac{\partial \boldsymbol{\Pi}^{ij}}{\partial \boldsymbol{\Phi}^{pq}} = \frac{\delta_{pi}\delta_{qj} \exp \boldsymbol{\Phi}^{ij} \left[\sum_k \exp \boldsymbol{\Phi}^{kj}\right] - \delta_{qj} \exp \boldsymbol{\Phi}^{ij} \exp \boldsymbol{\Phi}^{pj}}{\left[\sum_k \exp \boldsymbol{\Phi}^{kj}\right]^2}\,, \tag{3.7.9}$$

skąd

$$\frac{1}{\boldsymbol{\Pi}^{ij}} \frac{\partial \boldsymbol{\Pi}^{ij}}{\partial \boldsymbol{\Phi}^{pq}} = \delta_{pi}\delta_{qj} - \delta_{qj}\boldsymbol{\Pi}^{pj}\,. \tag{3.7.10}$$

Z powyższego zatem

$$\frac{\partial L}{\partial \boldsymbol{\Phi}^{pq}} = -\sum_{i,j} \boldsymbol{T}^{ij}\delta_{pi}\delta_{qj} + \sum_{i,j} \boldsymbol{T}^{ij}\delta_{qj}\boldsymbol{\Pi}^{pj} = \boldsymbol{\Pi}^{pq} - \boldsymbol{T}^{pq}\,, \tag{3.7.11}$$

gdzie skorzystaliśmy z faktu, iż z konstrukcji dla dowolnej kolumny q macierzy $\boldsymbol{T}$ zachodzi $\sum_i \boldsymbol{T}^{iq} = 1$. Możemy zapisać powyższy wzór w eleganckiej postaci macierzowej

$$\boxed{\frac{\partial L}{\partial \boldsymbol{\Phi}} = \boldsymbol{\Pi} - \boldsymbol{T}\,.} \tag{3.7.12}$$

Pochodną funkcji kosztu po parametrach $\boldsymbol{W}$ możemy zatem obliczyć jako

$$\frac{\partial L}{\partial \boldsymbol{W}^{rs}} = \sum_{p,q} \frac{\partial L}{\partial \boldsymbol{\Phi}^{pq}} \frac{\partial \boldsymbol{\Phi}^{pq}}{\partial \boldsymbol{W}^{rs}}\,, \tag{3.7.13}$$

gdzie

$$\frac{\partial \boldsymbol{\Phi}^{pq}}{\partial \boldsymbol{W}^{rs}} = \sum_t \delta_{rp}\delta_{st}\boldsymbol{X}^{tq} = \delta_{rp}\boldsymbol{X}^{sq}\,, \tag{3.7.14}$$

skąd

$$\frac{\partial L}{\partial \boldsymbol{W}^{rs}} = \sum_q (\boldsymbol{\Pi}^{rq} - \boldsymbol{T}^{rq})\boldsymbol{X}^{sq}\,, \tag{3.7.15}$$

co również możemy zapisać w zwartej postaci macierzowej

$$\boxed{\frac{\partial L}{\partial \boldsymbol{W}} = (\boldsymbol{\Pi} - \boldsymbol{T})\boldsymbol{X}^T\,.} \tag{3.7.16}$$