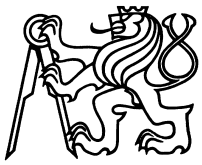
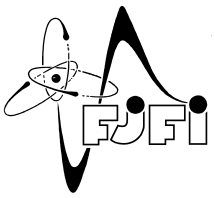


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



Klasifikace dat popsaných stromovou strukturou

Classification of tree–structured data

Bakalářská práce

Autor: **Lukáš Kulička**

Vedoucí práce: **doc. Ing. Václav Šmídl, Ph.D.**

Konzultant: **doc. Ing. Tomáš Pevný, Ph.D.**

Akademický rok: 2019/2020

České vysoké učení technické v Praze, Fakulta jaderná a fyzikálně inženýrská

Katedra: matematiky

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Lukáš Kulička
Studijní program:	Aplikace přírodních věd
Obor:	Matematické inženýrství
Zaměření:	Aplikované matematicko-stochastické metody
Název práce (česky):	Klasifikace dat popsaných stromovou strukturou
Název práce (anglicky):	Classification of tree-structured data

Pokyny pro vypracování:

1. Seznamte se se základními metodami klasifikace dat popsaných vektorem příznaků. Zvláštní pozornost věnujte metodám založeným na flexibilních parametrizacích pomocí neuronových sítí. Demonstrujte principy metod na jednoduchých příkladech.
2. Seznamte se s popisem dat pomocí stromové struktury. Zvláštní pozornost věnujte metodám více instančního učení (multiple instance learning). Seznamte se s konceptem vnořeného prostoru (embedded space) a jeho reprezentace pomocí neuronových sítí.
3. Navrhněte několik příkladů typů dat se stromovou strukturou a pro každý z nich vytvořte klasifikační úlohu. Navrhněte algoritmus jejího řešení a diskutujte vhodnost jednotlivých architektur neuronových sítí.
4. Seznamte se s metodou učení řídkých reprezentací pomocí regularizace neboli apriorního rozložení parametrů. Zvláštní pozornost věnujte principu automatic relevance determination. Aplikujte tuto metodu na jednoduché klasifikační úlohy z předchozího kroku. Diskutujte výsledné odhady.
5. Vyvinutou metodu aplikujte na vhodně zvolená reálná data a diskutujte vliv řídkosti řešení na výsledné odhady.

Doporučená literatura:

1. C. M. Bishop, Pattern recognition and machine learning. Springer, 2006.
2. T. Pevny, P. Somol, Discriminative models for multi-instance problems with tree structure. In 'Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security', ACM, 2016, 83–91.
3. J. Stiborek, T. Pevný, M. Rehák, Multiple instance learning for malware classification. Expert Systems with Applications 93, 2018, 346–357.

Jméno a pracoviště vedoucího bakalářské práce:

Doc. Ing. Václav Šmídl, PhD.
ÚTIA AV ČR, Pod vodárenskou věží 4, 180 00 Praha 8

Jméno a pracoviště konzultanta:

Doc. Ing. Tomáš Pevný, Ph.D.
Katedra počítačů, FEL ČVUT PRAHA, Karlovo nám. 13, 121 35 Praha 2

Datum zadání bakalářské práce: 31.10.2019

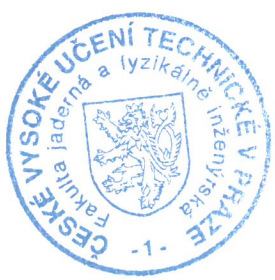
Datum odevzdání bakalářské práce: 7.7.2020

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 23. října 2019

.....
garant oboru

.....
vedoucí katedry



.....
děkan

Poděkování:
Rád bych zde poděkoval především svému školiteli doc. Ing. Václavu Šmídlovi, Ph.D. za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé bakalářské práce. Bylo mi ctí a výsadou pracovat a vzdělávat se po jeho boku. Dále děkuji svému konzultantovi doc. Ing. Tomáši Pevnému, Ph.D. za podnětné návrhy k práci a korekci. V neposlední řadě bych rád poděkoval svým rodičům Simoně Kuličkové a Mgr. Jiřímu Kuličkovi, Ph.D. za toleranci a podporu během celého bakalářského studia.

Čestné prohlášení:
Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 24. července 2020 Lukáš Kulička

Název práce:

Klasifikace dat popsaných stromovou strukturou

Autor: Lukáš Kulička

Obor: Matematické inženýrství

Zaměření: Aplikované matematicko–stochastické metody

Druh práce: Bakalářská práce

Vedoucí práce: doc. Ing. Václav Šmídl, Ph.D.
ÚTIA AV ČR, Pod vodárenskou věží 4, 180 00 Praha 8

Konzultant: doc. Ing. Tomáš Pevný, Ph.D.
Katedra počítačů, FEL ČVUT Praha, Karlovo nám. 13, 121 35 Praha 2

Abstrakt: Bakalářská práce se zabývá klasifikací dat popsaných stromovou strukturou. V rámci práce je uvedeno a důkladně popsáno široké spektrum různých matematických disciplín a teorie optimalizace, které je posléze aplikováno na demonstrativní příklady. Odvozen je též mocný optimalizační nástroj v oblasti bayesovského strojového učení jménem Evidence Lower Bound. Předvedeny jsou neuronové sítě a jejich souvislost s logistickou regresí a klasifikačními úlohami. Velký důraz je kladen na uchopení pojmu vysvětlitelné umělé inteligence a reprezentace modelů pomocí řídkých parametrizací. Zejména na metodu Automatic Relevance Determination a její význam. Závěrem jsou popsány koncept vnořeného prostoru a v současnosti dynamicky rozvíjející se hierarchické více–instanční učení včetně jeho spojení se stromovými strukturami.

Klíčová slova: klasifikace, řídkost, stromová struktura, více–instanční učení

Title:

Classification of tree–structured data

Author: Lukáš Kulička

Abstract: The bachelor thesis deals with the classification of tree–structured data. The thesis presents and thoroughly describes a wide range of the different mathematical disciplines and optimization theory, which is then applied to the demonstrative examples. A powerful optimization tool in Bayesian machine learning called Evidence Lower Bond is also derived. Neural networks and their connection with logistic regression and classification tasks are presented. Great emphasis is placed on grasping of concept of explainable artificial intelligence and model representation using sparse parameterizations. Especially on the Automatic Relevance Determination method and its significance. The concept of embedded space and the currently dynamically developing hierarchical multi–instance learning, including its connection with tree structures, are finally described.

Key words: classification, multiple–instance learning, sparsity, tree structure

Obsah

Seznam obrázků

Značení

Symbol	Význam
Ω	množina elementárních jevů
A, B, C	množiny
$\emptyset$	prázdná množina
$\mathcal{A}, \mathcal{B}, C$	systémy množin
$\cap$,$\cup$	průnik, sjednocení množin
$\mathbb{A}, \mathbb{B}, \mathbb{C}$	matice
$\mathbb{I}$	jednotková matice
$\mathbf{\Sigma}$	kovarianční matice
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	vektory
$\mathsf{A}, \mathsf{B}, \mathsf{C}$	funkcionály
$\mathscr{A}, \mathscr{B}, \mathscr{C}$	prostory
$\mathbf{N}$	množina přirozených čísel
$\mathbf{N}_0$	$\mathbf{N} \cup \{0\}$
$\mathbf{R}$	množina reálných čísel
$\hat{n}$	$\{m \in \mathbf{N} : m \leq n\}$
X	jednorozměrná náhodná veličina
$\mathbf{X}$	d–rozměrná náhodná veličina
$\mathrm{X} \sim$	jednorozměrná náhodná veličina se řídí rozdělením
$\boldsymbol{\theta}$	vektor parametrů nebo vah
X	množina vektorů $\mathbf{x}_i$
$\mathbf{x}_i$	i–té pozorování
$\mathbf{x}_{ij}$	j–tá proměnná i–tého pozorování
$\propto$	úměrně
$\mathrm{Tr}\,(\mathbb{A})$	stopa matice $\mathbb{A}$
$\mathrm{supp}\, p$	nosič funkce p
x	proměnná x
$\mathbf{x}$	d–dimenzionální vektor proměnných
$f(x)$	funkce jedné proměnné, $f(x) : \mathbf{R} \to \mathbf{R}$
$f(\mathbf{x})$	funkce d proměnných, $f(\mathbf{x}) : \mathbf{R}^d \to \mathbf{R}$
$\mathbf{f}(\mathbf{x})$	l–dimenzionální vektorová funkce d proměnných, $\mathbf{f}(\mathbf{x}) : \mathbf{R}^d \to \mathbf{R}^l$
$\Theta(x)$	Heavisideova funkce
$\delta(x)$	Diracova δ–funkce
$\mathrm{W}(X)$	množinová funkce $\mathrm{W}(X) : X \to \mathbf{R}$
$\mathcal{I}(X)$	charakteristická funkce $\mathcal{I}(X) : X \to \{0, 1\}$
ozn.	označíme

Úvod

Žijeme v době, kdy jsou na zpracování a interpretaci dat kladeny stále vyšší a vyšší nároky. Moderní technologie jsou schopny měřit a zaznamenávat obrovská množství dat (tzv. big data), která jsou ovšem potřeba náležitě zpracovat a vyhodnotit. K tomu existuje mnoho mocných a silných nástrojů z oblasti umělé inteligence (AI), jako například hluboké neuronové sítě, Bayesovské sítě, Support Vector Machines nebo rozhodovací stromy. Pokud jsou data trénována dostatečně dlouho a je k dispozici odpovídající výpočetně schopný hardware, není problém model natrénovat i s vysokou přesností. Drtivá většina z nich však k učení používá tzv. black box, který si lze představit jako záhadnou skříňku, do které se naměřená data vloží a vypadne požadovaný výsledek. Co se děje v něm a v rozhodovacích procesech napříč modelem, zůstává často záhadou. Další obtíží může být náročnost modelů na optimalizaci. Hlavně pokud obsahují statisíce parametrů, z nichž jen některé mohou být pro model důležité. Proto je v poslední době kladen větší důraz na tzv. vysvětlitelnost modelu pomocí řídké parametrizace, která činí jeho rozhodovací procesy více pochopitelnými pro lidi za současného výrazného snížení výpočetní náročnosti díky absenci nevýznamných parametrů. Též se zjišťuje, že popisovat model pomocí vektorů příznaků hierarchicky srovnaných do podoby stromové struktury je přirozenější a lépe interpretovatelné než popis klasickými vektory.

V úvodní části práce bude vypracován podrobný přehled důležitých definic a teorému z oblasti teorie míry, pravděpodobnosti, matematické statistiky a teorie grafů. Sekundovat mu bude seznámení se se základními pojmy z oblasti optimalizace, zejména s pojmem Gradient Descent a jeho významem v procesu učení. Pomocí Bayesova teorému a nástroje Elbo bude možno nahlížet na úlohy strojového učení pravděpodobnostním pohledem. Proto bude ukázáno srovnání, kde budou vybrané úlohy řešeny jak statistickým, tak pravděpodobnostním přístupem. V neposlední řadě bude kladen velký důraz na porozumění stromovým strukturám a jejich reprezentace pomocí grafů.

Text poté přechází na praktické příklady, ve kterých bude aplikována vybudovaná teorie v kombinaci s numerickým výpočtem a grafickým znázorněním. Dále bude představena metoda logistické regrese, jež tvoří základ řešení vícetřídových klasifikačních úloh a její souvislost s neuronovou sítí.

Nakonec budou předvedeny metody, jakými lze docílit řídkých reprezentací modelu pomocí odpovídajících apriorních rozdělení. Větší důraz bude kladen na metodu Automatic Relevance Determination. Závěrem bude zmíněn pojem více–instančního učení (MIL), který rozšíří možnosti popisu modelu pomocí příznaků a bude definován vnořený prostor (embedded space) spolu s jeho reprezentací pomocí neuronových sítí. Práce bude zakončena náhledem do hierarchického více–instančního učení (HMill).

Primárním cílem této práce bude seznámit se s teorií a osvojit si ji k použití na řešení jednoduchých demonstrativních příkladů klasifikačních a regresních úloh především pomocí řídkých parametrizací až po jejich reprezentace stromovými strukturami. K jejich numerickému řešení bude výhradně použit programovací jazyk Julia ve verzi 1.4.

Tato bakalářská práce představuje předstupeň hlubšího porozumění složitějším modelům a metodám. Především potenciálu hierarchického pohledu na stromové struktury dat a jejich řídkých reprezentací právě pomocí MIL. Proto na ni bude navázáno další akademickou prací.

Kapitola 1

Základní teorie

1.1 Teorie pravděpodobnosti

Cílem sekce 1.1 bude řádně zadefinovat základní pojmy z teorie míry a pravděpodobnosti potřebné k vyslovení a dokázání Bayesova teorému, jenž pro nás bude nesmírně důležitý. K tomu je především potřeba seznámit se s definicí *pravděpodobnostního prostoru*, na kterém se budeme pohybovat. Definice a teorémy jsou převzaty z [1, 6, 7] a upraveny podle odpovídajícího značení.

1.1.1 Pravděpodobnostní prostor a Bayesův teorém

Definice 1.1.1. Definujeme následující pojmy:

1. **Elementární jev** ω je jev, jehož vnitřní strukturu již dále nerozlišujeme.
2. **Základní množinou** Ω nazveme množinu všech elementárních jevů ω.
3. **Jev** A je buď elementárním jevem ω, nebo je libovolnou podmnožinou základní množiny, tj. $A \subset \Omega$.
4. Řekneme, že jev A **nastal**, pokud nastal elementární jev $\omega \in \Omega$ a navíc $\omega \in A$.
5. Definujeme **komplementární jev** A^c k A jako $\omega \in A^c \Leftrightarrow \omega \notin A$.

Definice 1.1.2. Nechť je dána základní množina Ω a nějaký její systém podmnožin $\mathcal{A}$, tj. $\mathcal{A} \subset \mathcal{P}(\Omega)$ [1] splňující následující axiomy:

1. $\Omega \in \mathcal{A}$,
2. $(\forall A \in \mathcal{A})(A^c \in \mathcal{A})$,
3. $(\forall j \in \mathbf{N})(A_j \in \mathcal{A})(\bigcup_{j=1}^{+\infty} A_j \in \mathcal{A})$.

Pak $\mathcal{A}$ nazýváme σ**–algebrou** jevů (spočetně mnoha množin) na základní množině Ω.

[1] $\mathcal{P}(\Omega)$ značí potenci množiny Ω

Definice 1.1.3. Nechť je dána neprázdná základní množina Ω a na ní σ–algebra $\mathcal{A}$. Pak libovolnou funkci $\mathrm{P} : \mathcal{A} \to \mathbf{R}$ splňující tři Kolmogorovovy[2] axiomy:

1. $\mathrm{P}(\Omega) = 1$,
2. $(\forall A \in \mathcal{A})(\mathrm{P}(A) \geq 0)$,
3. $\left(\forall (A_j)_{j=1}^{+\infty} \in \mathcal{A}\right) \left(\mathrm{P}(\sum_{j=1}^{+\infty} A_j) = \sum_{j=1}^{+\infty} \mathrm{P}(A_j)\right)$, kde A_j jsou navzájem disjuktní množiny,

nazveme **pravděpodobnostní mírou**.

Definice 1.1.4. Trojici tvořenou základní množinou Ω, σ–algebrou $\mathcal{A}$ a pravděpodobnostní mírou P nazveme **pravděpodobnostním prostorem** a značíme $(\Omega, \mathcal{A}, \mathrm{P})$.

Nyní máme k dispozici prostor vybavený pravděpodobnostní mírou P, na kterém dokážeme určitým jevům přirazovat jejich odpovídající pravděpodobnosti. Díky tomu dokážeme zadefinovat následující vztahy:

Definice 1.1.5. Nechť jsou dány jevy $A, B \in \mathcal{A}$, $\mathrm{P}(B) > 0$. Pak definujeme **podmíněnou pravděpodobnost** vztahem

$$\mathrm{P}(A|B) := \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)}, \tag{1.1}$$

čímž rozumíme pravděpodobnost jevu A za předpokladu, že nastal jev B.

Teorém 1.1.6. (Součinové pravidlo) Nechť $A_1, \ldots, A_n \in \mathcal{A}$ jsou takové, že $\mathrm{P}(A_1 \cdot \ldots \cdot A_n) > 0$, kde součiny jsou ve smyslu průniků. Pak $\forall n \in \mathbf{N}$ platí

$$\mathrm{P}(A_1 \cdot \ldots \cdot A_n) = \mathrm{P}(A_1) \cdot \mathrm{P}(A_2|A_1) \cdot \mathrm{P}(A_3|A_1 \cdot A_2) \cdot \ldots \cdot \mathrm{P}(A_n|A_1 \cdot \ldots \cdot A_{n-1}). \tag{1.2}$$

Definice 1.1.7. Nechť je dán pravděpodobnostní prostor $(\Omega, \mathcal{A}, \mathrm{P})$. Systém jevů $\mathcal{H} = \{H_j | j \in \mathbf{N}\}$ tvoří **úplný rozklad množiny** Ω, pokud

- jevy z $\mathcal{H}$ jsou vzájemně neslučitelné, tj. $A_i \cap A_j = \emptyset$, $\forall i, j \in \mathbf{N}$,
- $(\forall j \in \mathbf{N})(\mathrm{P}(H_j) > 0)$,
- platí $\mathrm{P}(\sum_{j=1}^{n,+\infty} H_j) = 1$. To znamená, že při pokrývání základní množiny Ω můžeme vynechat některé množiny. Ty však musí být nulové míry. Symbol $(n, +\infty)$ značí konečnost, nebo spočetnost.

Teorém 1.1.8. (Součtové pravidlo) Nechť systém jevů $\{H_j | j \in \mathbf{N}\}$ tvoří úplný rozklad základní množiny Ω na $(\Omega, \mathcal{A}, \mathrm{P})$ a $A \in \mathcal{A}$. Potom platí:

$$\mathrm{P}(A) = \sum_{j=1}^{n,+\infty} \mathrm{P}(A|H_j) \cdot \mathrm{P}(H_j). \tag{1.3}$$

Teorém 1.1.9. (Symetrie) Nechť je dán $(\Omega, \mathcal{A}, \mathrm{P})$ a A,$B \in \mathcal{A}$. Pak platí:

$$\mathrm{P}(A \cap B) = \mathrm{P}(B \cap A) \tag{1.4}$$

[2] *Andrej Nikolajevič Kolmogorov* (1903–1987)

Na pravděpodobnostním prostoru s využitím předchozích definic a teorémů vyslovíme a dokážeme jeden z nejdůležitějších teorémů, který nám později poskytne silný nástroj, jak alternativně řešit úlohy s využitím pravděpodobností.

Teorém 1.1.10. (Bayesův[3]) Nechť je dán $(\Omega, \mathcal{A}, \mathrm{P})$, $A \in \mathcal{A}$ a $\{B_j | j \in \mathbf{N}\}$ tvoří úplný rozklad základní množiny Ω. Pak platí:

$$\mathrm{P}(B_k|A) = \frac{\mathrm{P}(A|B_k) \cdot \mathrm{P}(B_k)}{\sum_{j=1}^{n,+\infty} \mathrm{P}(A|B_j) \cdot \mathrm{P}(B_j)}. \tag{1.5}$$

Důkaz.

$$\mathrm{P}(B_k|A) \overset{1.1}{=} \frac{\mathrm{P}(B_k \cap A)}{\mathrm{P}(A)} \overset{1.3}{=} \frac{\mathrm{P}(B_k \cap A)}{\sum_{j=1}^{n,+\infty} \mathrm{P}(A|B_j) \cdot \mathrm{P}(B_j)} \overset{1.4}{=} \frac{\mathrm{P}(A \cap B_k)}{\sum_{j=1}^{n,+\infty} \mathrm{P}(A|B_j) \cdot \mathrm{P}(B_j)} \overset{1.2}{=} \frac{\mathrm{P}(A|B_k) \cdot \mathrm{P}(B_k)}{\sum_{j=1}^{n,+\infty} \mathrm{P}(A|B_j) \cdot \mathrm{P}(B_j)}$$

□

Než se přesuneme na prostor reálných čísel z našeho velice abstraktního prostoru jevů, je vhodné si uvědomit, co Bayesův teorém říká. Doslova obrací chod času. Podívejme se dále, jaký význam zde má jmenovatel:

$$\sum_{j=1}^{n,+\infty} \mathrm{P}(A|B_j) \cdot \mathrm{P}(B_j) = \mathrm{P}(A). \tag{1.6}$$

Lze vidět, že výraz (1.6) hraje roli normalizační konstanty, a proto bude pro naše nadcházející účely vhodné vyslovený Bayesův teorém trochu přeformulovat. Pouze výjimečně se v úlohách optimalizace a numerického počítání setkáme se zcela přesným řešením úlohy. Mnohdy hledáme pouze určité druhy závislostí mezi vstupy a výstupy. Zavedeme proto symbol $\propto$ jakožto *úměrný*.

Teorém 1.1.11. (Bayesův, alternativní) Nechť platí předpoklady (1.1.10). Pokud nás zajímají pouze závislosti pravděpodobností, či nejsou konstanty pro účel výpočtu důležité, pak lze (1.5) rozšířit na

$$\mathrm{P}(B_k|A) = \frac{\mathrm{P}(A|B_k) \cdot \mathrm{P}(B_k)}{\sum_{j=1}^{n,+\infty} \mathrm{P}(A|B_j) \cdot \mathrm{P}(B_j)} \propto \mathrm{P}(A|B_k) \cdot \mathrm{P}(B_k) \tag{1.7}$$

1.1.2 Hustota pravděpodobnosti náhodné veličiny

Již jsme formálně zadefinovali pravděpodobnostní prostor a Bayesův teorém. Nyní je důležité přejít na méně abstraktní prostor, a to prostor hustot pravděpodobností jakožto funkcí zobrazujících do reálných čísel a zavést známé integrální charakteristiky. Odvození a formální definice spojité náhodné veličiny je nad rámec této práce a pro naše účely zbytečná.

Definice 1.1.12. Nechť je dán prostor $(\Omega, \mathcal{A}, \mathrm{P})$. **Náhodnou veličinou X** nazveme každé měřitelné zobrazení $\mathbf{X} : \Omega \to \mathbf{R}^d$.

[3] *Thomas Bayes* (1701–1761)

Definice 1.1.13. Nechť $\mathbf{X} = (\mathrm{X}_1, \ldots, \mathrm{X}_d)$ je náhodná veličina na prostoru $(\Omega, \mathcal{A}, \mathrm{P})$. Náhodnou veličinu $\mathbf{X}$ nazveme **diskrétní** právě tehdy, když nabývá nejvýše spočetně mnoha hodnot. Její **hustotu pravděpodobnosti** definujeme vztahem

$$p_{\mathbf{X}}(\mathbf{x}) = \begin{cases} \mathrm{P}(\mathbf{X} = \mathbf{x}_k), & \text{pro } \mathbf{x} = \mathbf{x}_k \\ 0, & \text{jinak.} \end{cases}$$

Dále definujeme její **distribuční funkci** (pro zjednodušení uvažujeme pouze $d = 1$) jako sumu přes všechny nabyté hodnoty, tj. $F_X(x) = \mathrm{P}(X \leq x) = \sum_{x_k \leq x} \mathrm{P}(X = x_k)$.
Povšimněme si též *normovanosti* hustoty pravděpodobnosti na jedničku: $\sum_{k=1}^{n,+\infty} p_X(x_k) = 1$. Což formálně odpovídá definici (1.1.3). Platí také, že $(\forall k \in \mathbf{N})(p_X(x_k) \in [0, 1])$.

Definice 1.1.14. Nechť $\mathbf{X} = (\mathrm{X}_1, \ldots, \mathrm{X}_d)$ je náhodná veličina na prostoru $(\Omega, \mathcal{A}, \mathrm{P})$. Náhodnou veličinu $\mathbf{X}$ nazveme **spojitou**, pokud je její obor hodnot souvislá podmnožina $\mathbf{R}^d$.
Její **distribuční funkcí** v bodě $\mathbf{x} \in \mathbf{R}^d$ rozumíme

$$F_{\mathbf{X}}(\mathbf{x}) = \mathrm{P}(\mathbf{X} \leq \mathbf{x}) = \int_{-\infty}^{x_1} \ldots \int_{-\infty}^{x_d} p_{\mathbf{X}}(\mathbf{t}) d\mathbf{t}, \tag{1.8}$$

kde $p_{\mathbf{X}}(\mathbf{x})$ představuje **hustotu pravděpodobnosti**. Pro jednorozměrný, respektive d–rozměrný případ budeme používat značení $p(x)$, respektive $p(\mathbf{x})$, kde $\mathbf{x}$ značí realizaci d–rozměrné náhodné veličiny $\mathbf{X}$.
Také platí (podobně jako u diskrétní veličiny) $\int_{\mathbf{R}^d} p(\mathbf{x}) d\mathbf{x} = 1$ a z (1.8) plyne $p(\mathbf{x}) \geq 0$.

Teorém 1.1.15. (Nezávislost) Nechť $\mathbf{X} = (\mathrm{X}_1, \ldots, \mathrm{X}_d)$ je spojitá náhodná veličina na $(\Omega, \mathcal{A}, \mathrm{P})$. Řekneme, že $\mathrm{X}_1, \ldots, \mathrm{X}_d$ jsou **nezávislé** právě tehdy, když

$$\Big(\forall \mathbf{x} \in \mathbf{R}^d\Big)\left(p_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{d} p_{\mathrm{X}_i}(x_i)\right) \tag{1.9}$$

Teorém 1.1.16. (Marginální hustota) Nechť $\mathbf{X} = (\mathrm{X}_1, \ldots, \mathrm{X}_d)$ je spojitá náhodná veličina na $(\Omega, \mathcal{A}, \mathrm{P})$. Označme $\mathbf{X'} = (\mathrm{X}_1, \ldots, \mathrm{X}_{j-1}, \mathrm{X}_{j+1}, \ldots, \mathrm{X}_d)$. Pak $\mathbf{X'}$ je spojitá náhodná veličina a pro její hustotu platí

$$\Big(\forall \mathbf{x'} \in \mathbf{R}^{d-1}\Big)\left(p_{\mathbf{X'}}(\mathbf{x'}) = \int_{\mathbf{R}} p_{\mathbf{X}}(\mathbf{x}) dx_j\right). \tag{1.10}$$

Funkci $p_{\mathbf{X'}}(\mathbf{x'})$ nazveme **marginální hustotou pravděpodobnosti**.

1.1.3 Integrální charakteristiky

Poté, co jsme zavedli diskrétní a spojité náhodné veličiny, jejich distribuční funkce a hustoty pravděpodobností, jsme oprávněni zadefinovat známé charakteristiky. Využijeme též faktu, že složením náhodné vektorové veličiny $\mathbf{X}$ s měřitelnou vektorovou funkcí $\mathbf{g}$, tj. $\mathbf{g} \circ \mathbf{X} = \mathbf{g(X)}$, vznikne opět náhodná vektorová veličina, což nám rozšíří klasicky zavedené definice.

Abychom dodrželi konzistenci značení s literaturou [1], nebudeme používat označení náhodné veličiny klasickým, pravděpodobnostním způsobem, ale ztotožníme se s oborovou konvencí, ve které jsou náhodné veličiny označené velkým $\mathbf{X}$ (včetně jejich složenin s různými měřitelnými vektorovými funkcemi) označovány malými písmeny jakožto jejich realizacemi. Můžeme je nazvat jednoduše *funkcemi*.

V tomto duchu zavedeme další užitečné pojmy a definice.

Definice 1.1.17. (Support hustoty pravděpodobnosti) **Supportem** (též nosičem) **hustoty pravděpodobnosti** $p(x)$ nazveme uzávěr množiny nenulových obrazů funkce $p(x)$, tj. supp $p = \overline{\{x \in \mathbf{R} | p(x) > 0\}}$.
U všech nadcházejících určitých integrálů, pokud nebude uvedeno jinak, budeme vždy integrovat přes celý support hustoty $p(x)$.

Poznámka. Tento pojem se dá samozřejmě rozšířit i na prostor $\mathbf{R}^d$: supp $p = \overline{\{\mathbf{x} \in \mathbf{R}^d | p(\mathbf{x}) > 0\}}$, kde $p(\mathbf{x})$ je funkce d proměnných.

Definice 1.1.18. (Střední hodnota) Nechť $f(x)$ je měřitelná integrabilní reálná funkce reálné proměnné. Průměrnou hodnotu funkce $f(x)$ váženou pravděpodobnostní distribucí $p(x)$ nazveme **střední** (též očekávanou) **hodnotou** funkce $f(x)$ vzhledem k $p(x)$. Značíme:

- v diskrétním případě:
$$\mathsf{E}_{p(x)}[f(x)] := \sum_x p(x)f(x) \overset{ozn.}{=} \mathsf{E}[f] \tag{1.11}$$
- ve spojitém případě:
$$\mathsf{E}_{p(x)}[f(x)] := \int p(x)f(x)dx \overset{ozn.}{=} \mathsf{E}[f], \tag{1.12}$$

kde sčítáme přes všechny nabyté hodnoty a integrujeme přes celý support (1.1.17) hustoty pravděpodobnosti.

Definice 1.1.19. (Podmíněná střední hodnota) **Podmíněnou střední hodnotou** funkce $f(x)$ za podmínky y rozumíme

- v diskrétním případě:
$$\mathsf{E}_x[f|y] = \sum_x p(x|y)f(x) \tag{1.13}$$
- ve spojitém případě:
$$\mathsf{E}_x[f|y] = \int p(x|y)f(x)dx. \tag{1.14}$$

Definice 1.1.20. (Rozptyl) Nechť $f(x)$ je měřitelná reálná funkce reálné proměnné integrabilní s kvadrátem a existuje její střední hodnota. **Rozptylem** funkce $f(x)$ rozumíme vztah

$$\mathsf{D}[f(x)] := \mathsf{E}\left[(f(x) - \mathsf{E}[f(x)])^2\right] \overset{ozn.}{=} \mathrm{var}[f]. \tag{1.15}$$

Rozptyl tedy vyjadřuje míru variability funkce $f(x)$ kolem její střední hodnoty.

Definice 1.1.21. (Kovariance) Nechť jsou dány funkce $f(x)$ a $g(y)$. **Kovariancí** těchto dvou funkcí rozumíme číslo

$$\mathrm{cov}[f(x), g(y)] := \mathsf{E}[(f(x) - \mathsf{E}[f(x)])(g(y) - \mathsf{E}[g(y)])] \tag{1.16}$$

Kovariance ukazuje míru lineární závislosti mezi dvěma funkcemi. Pokud, bez újmy na obecnosti, použijeme identické funkce, tj. $f(x) = x$ a $g(y) = y$ a roznásobíme, dostaneme známý vztah:

$$\mathrm{cov}[x, y] = \mathsf{E}[xy] - \mathsf{E}[x]\mathsf{E}[y]. \tag{1.17}$$

Nesetkáme se vždy pouze s jednorozměrným případem. Ba naopak, v našem budoucím snažení to bude spíše rarita. Je tedy určitě na místě definovat pojem **kovarianční matice** vektorové funkce, který je v analýze vícerozměrných hustot pravděpodobností a potřebných výpočtech velice důležitý. Pro zjednodušení použijeme v následující definici identickou vektorovou funkci $\mathbf{f(x)} = \mathbf{x}$.

Definice 1.1.22. (Vektorová střední hodnota) Nechť je dána identická vektorová funkce **f(x)** = **x**. **Vektorovou střední hodnotu** definujeme jako vektor středních hodnot jednotlivých složek vektorové funkce

$$\mathsf{E}[\mathbf{x}] := (\mathsf{E}[x_1], \ldots, \mathsf{E}[x_d]). \tag{1.18}$$

Definice 1.1.23. (Kovarianční matice) Nechť je dána identická vektorová funkce **f(x)** = **x**. Matici o rozměrech $d \times d$ definovanou vztahem

$$\mathbb{C}(\mathbf{x}) := \left(\operatorname{cov}\left[x_i, x_j\right]\right)_{i,j=1}^{d} \overset{ozn.}{=} \mathbf{\Sigma(x)} \tag{1.19}$$

nazveme **kovarianční maticí** identické vektorové funkce.

Rozšiřme nyní vztah (1.17) na d–rozměrné náhodné vektory. Uvažujme opět identické vektorové funkce **f(x)** = **x** = $(x_1, \ldots, x_d)$ a **g(y)** = **y** = $(y_1, \ldots, y_d)$. Víme, že výstupem kovariance musí být skalár.

$$\operatorname{cov}[\mathbf{x}, \mathbf{y}] \overset{1.16}{=} \mathsf{E}\left[(\mathbf{x} - \mathsf{E}[\mathbf{x}])(\mathbf{y}^T - \mathsf{E}[\mathbf{y}^T])\right] \overset{1.17}{=} \mathsf{E}[\mathbf{x}\mathbf{y}^T] - \mathsf{E}[\mathbf{x}]\mathsf{E}[\mathbf{y}^T] \tag{1.20}$$

1.1.4 Kullback–Leiblerova divergence

Ne vždy je možné model nebo systém dokonale popsat pomocí známé pravděpodobnostní distribuce s odpovídající střední hodnotou a variancí. Je možné využít jiná pravděpodobnostní rozdělení, která známe, a posléze aplikovat Kullback–Leiblerovu[4] divergenci, jejíž výstup minimalizujeme pomocí k tomu určených numerických metod.

Definice 1.1.24. (Spojitá entropie) **Spojitou entropii** $\mathsf{H}[p]$ definujeme jako funkcionál, do jehož argumentu vkládáme pravděpodobnostní distribuci *p(x)*. Jeho výstupem nám bude informace o míře neuspořádanosti systému.

$$\mathsf{H}[p] := -\int p(x) \ln(p(x))\, dx \tag{1.21}$$

Definice 1.1.25. (Relativní entropie) Nechť $p(x)$ a $q(x)$ jsou pravděpodobnostní distribuce. **Relativní entropii** (též KL divergenci) definujeme vztahem

$$KL(p\|q) := -\int p(x) \ln\left\{\frac{q(x)}{p(x)}\right\} dx. \tag{1.22}$$

Jedná se o míru, která vyjadřuje, jak se jedna pravděpodobnostní distribuce liší od druhé.

Rozepsáním vztahu (1.22) vidíme, proč se *KL divergenci* přezdívá *relativní entropie*.

$$-\int p(x) \ln\left\{\frac{q(x)}{p(x)}\right\} dx = -\int p(x) \ln(q(x))\, dx - \left(-\int p(x) \ln(p(x))\, dx\right)$$

Z definice metriky je jasné, proč nemůže být KL divergence prohlášena za metriku, ačkoli se to intiutivně zcela nabízí. Obecně nesplňuje axiomy metriky číslo 2 a 3:

Ax. 1 : $KL(p\|q) \geq 0$ a $KL(p\|q) = 0 \Leftrightarrow p(x) = q(x)$

Ax. 2 : obecně platí: $KL(p\|q) \neq KL(q\|p)$

Ax. 3 : nesplňuje trojúhelníkovou nerovnost.

[4] *Solomon Kullback* (1907–1994), *Richard Leibler* (1914–2003)

1.1.5 Příklad spojitých pravděpodobnostních rozdělení

Nyní demonstrujme zadefinované pojmy, vyslovené teorémy a zavedenou symboliku na třech jednorozměrných spojitých a jednom vícerozměrném pravděpodobnostním rozdělení. Dílčími výpočty se zabývat nebudeme.

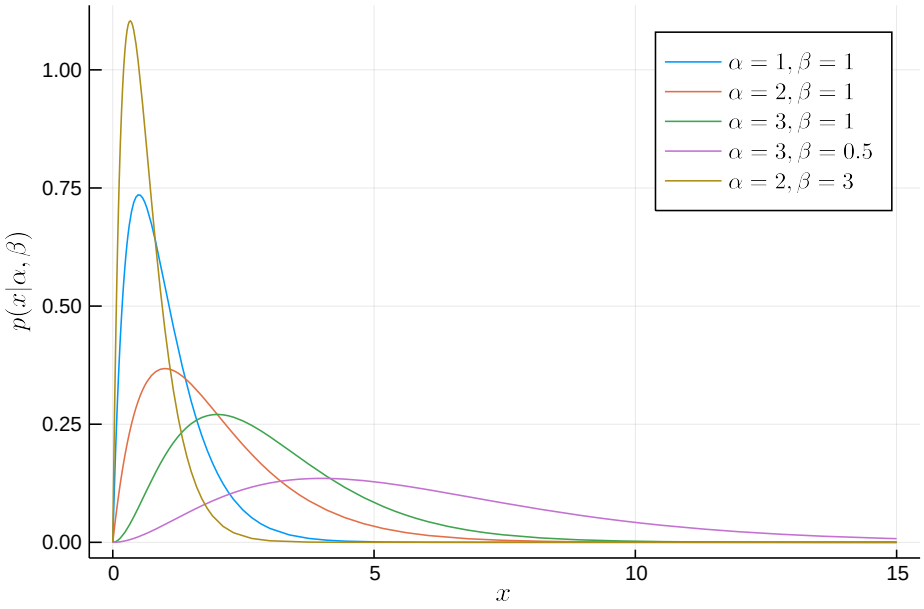
Γ rozdělení

Řekneme, že náhodná veličina X se řídí **Γ rozdělením** se dvěma parametry α a β, tj. $\mathrm{X} \sim \Gamma(\alpha,\beta)$, pokud odpovídající hustota pravděpodobnosti splňuje vztah:

$$p(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{\alpha-1}\exp(-\beta x). \tag{1.23}$$

- **Support hustoty**: $\operatorname{supp} p(x|\alpha,\beta) = \mathbf{R}^{+}$, kde $\alpha > 0$ a $\beta > 0$
- **Distribuční funkce**: $F_{\mathrm{X}}(x) \overset{1.8}{=} \int_0^x \frac{1}{\Gamma(\alpha)}\beta^{\alpha}t^{\alpha-1}\exp(-\beta t)dt = \frac{1}{\Gamma(\alpha)}\gamma(\alpha,\beta x)$[5]
- **Střední hodnota rozdělení**: $\mathsf{E}_{p(x)}[f(x)] \overset{1.12}{=} \int_0^{+\infty} x\frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{\alpha-1}\exp(-\beta x)dx = \frac{\alpha}{\beta}$
- **Variance rozdělení**: $\mathsf{D}[f(x)] \overset{1.15}{=} \mathsf{E}\left[(x-\frac{\alpha}{\beta})^2\right] = \frac{\alpha}{\beta^2}$
- **Entropie rozdělení**:

$$\mathsf{H}[p] \overset{1.21}{=} \int_0^{+\infty} \frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{\alpha-1}\exp(-\beta x)\left(\ln(\frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{\alpha-1}\exp(-\beta x))\right)dx =$$
$$= \alpha - \ln(\beta) + \ln(\Gamma(\alpha)) + (1-\alpha)\psi(\alpha)$$[6]



Obrázek 1.1: Příklad Γ rozdělení pro různé parametry α a β.

[5] $\gamma(\alpha,x) = \int_0^x t^{\alpha-1}\exp(-t)dt$ označována též jako *spodní neúplná* Γ–funkce
[6] $\psi(\alpha) = \frac{d}{d\alpha}\ln(\Gamma(\alpha))$ označována též jako *digamma funkce*

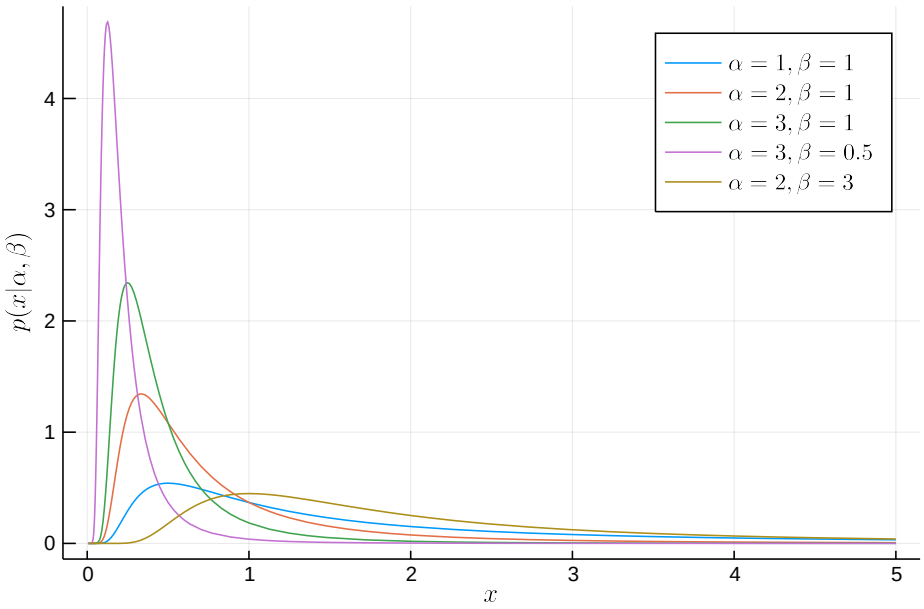
Inverzní Γ rozdělení

Řekneme, že náhodná veličina X se řídí **inverzním Γ rozdělením** se dvěma parametry α a β, tj. $X \sim \imath\Gamma(\alpha,\beta)$, pokud odpovídající hustota pravděpodobnosti splňuje vztah:

$$p(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{-\alpha-1}\exp\left(-\frac{\beta}{x}\right). \tag{1.24}$$

- **Support hustoty**: $\operatorname{supp} p(x|\alpha,\beta) = \mathbf{R}^{+}$, kde $\alpha > 0$ a $\beta > 0$
- **Distribuční funkce**: $F_X(x) \overset{1.8}{=} \int_0^x \frac{1}{\Gamma(\alpha)}\beta^{\alpha}t^{-\alpha-1}\exp\left(-\frac{\beta}{t}\right)dt = \frac{\Gamma(\alpha,\beta/x)}{\Gamma(\alpha)}$[7]
- **Střední hodnota rozdělení**: $\mathsf{E}_{p(x)}[f(x)] \overset{1.12}{=} \int_0^{+\infty} x\frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{-\alpha-1}\exp\left(-\frac{\beta}{x}\right)dx = \frac{\beta}{\alpha-1}$, pro $\alpha > 0$
- **Variance rozdělení**: $\mathsf{D}[f(x)] \overset{1.15}{=} \mathsf{E}\left[(x-\frac{\beta}{\alpha-1})^2\right] = \frac{\beta^2}{(\alpha-1)^2(\alpha-2)}$, pro $\alpha > 2$
- **Entropie rozdělení**:

$$\mathsf{H}[p] \overset{1.21}{=} \int_0^{+\infty} \frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{-\alpha-1}\exp\left(-\frac{\beta}{x}\right)\left(\ln(\frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{-\alpha-1}\exp\left(-\frac{\beta}{x}\right))\right)dx =$$
$$= \alpha + \ln(\beta\Gamma(\alpha)) - (1+\alpha)\psi(\alpha)$$



Obrázek 1.2: Příklad $\imath\Gamma$ rozdělení pro různé parametry α a β.

[7] $\Gamma(\alpha,x) = \int_x^{+\infty} t^{\alpha-1}\exp(-t)dt$ označována též jako *horní neúplná* Γ–funkce

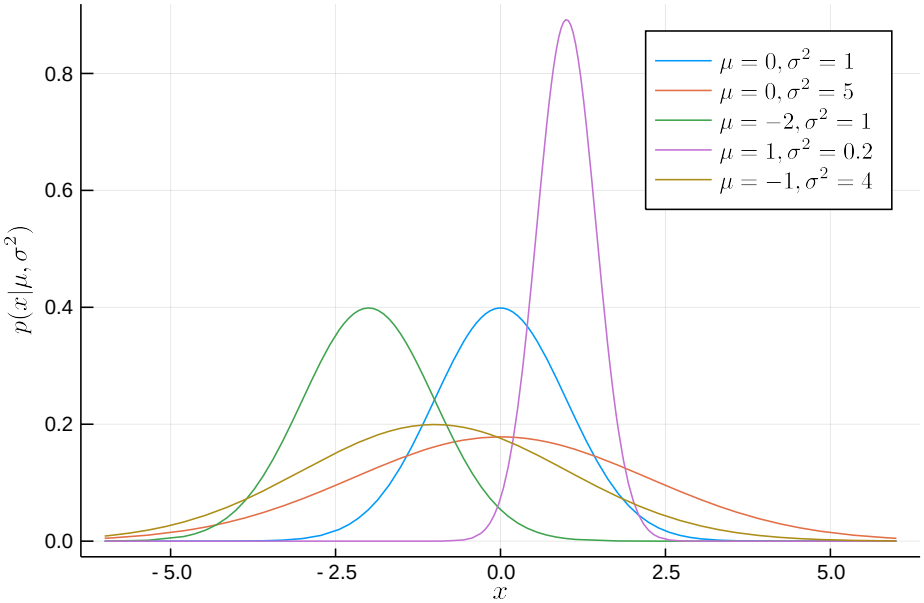
Gaussovo rozdělení

Řekneme, že náhodná veličina X se řídí **Gaussovým**[8] **rozdělením** se dvěma parametry μ a σ^2, tj. $\mathrm{X} \sim \mathcal{N}(\mu, \sigma^2)$, pokud odpovídající hustota pravděpodobnosti splňuje vztah:

$$p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \overset{ozn.}{=} \mathcal{N}(x|\mu,\sigma^2). \tag{1.25}$$

- **Support hustoty**: $\operatorname{supp} p(x|\mu,\sigma^2) = \mathbf{R}$, kde $\mu \in \mathbf{R}$ a $\sigma^2 > 0$
- **Distribuční funkce**: $F_{\mathrm{X}}(x) \overset{1.8}{=} \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(t-\mu)^2\right) dt = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right)\right)$[9]
- **Střední hodnota rozdělení**: $\mathsf{E}_{p(x)}[f(x)] \overset{1.12}{=} \int_{\mathbf{R}} x \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) dx = \mu$
- **Variance rozdělení**: $\mathsf{D}[f(x)] \overset{1.15}{=} \mathsf{E}\left[(x-\mu)^2\right] = \sigma^2$
- **Entropie rozdělení**:

$$\mathsf{H}[p] \overset{1.21}{=} \int_{\mathbf{R}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)\left(\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)\right)\right) dx =$$
$$= \frac{1}{2}\left(\ln(2\pi\sigma^2) + 1\right)$$



Obrázek 1.3: Příklad Gaussova rozdělení pro různé parametry μ a σ^2.

[8] *Carl Friedrich Gauss* (1777–1855)

[9] $\operatorname{erf} x = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} \exp(-t^2) dt$ označována též jako *Gaussova error funkce*

d–rozměrné Gaussovo rozdělení

Řekneme, že náhodná veličina $\mathbf{X} : \Omega \to \mathbf{R}^d$ se řídí **d–rozměrným Gaussovým rozdělením** s parametry $\boldsymbol{\mu}$ a $\boldsymbol{\Sigma}$, tj. $\mathbf{X} \sim \mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, pokud se každá lineární kombinace jejích d složek řídí jednorozměrným Gaussovým rozdělením. Odpovídající hustota pravděpodobnosti, jakožto funkce d proměnných, splňuje vztah:

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \overset{ozn.}{=} \mathcal{N}_d(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{1.26}$$

kde $d \in \mathbf{N}$, $\boldsymbol{\mu}$ je d–dimenzionální vektor středních hodnot (1.18), $\boldsymbol{\Sigma}$ kovarianční matice rozměrů $d \times d$ (1.19) a $|\boldsymbol{\Sigma}|$ označuje hodnotu determinantu matice $\boldsymbol{\Sigma}$.

1.1.6 Konvence značení

Dále je třeba jednoznačně určit, jaké značení (zejména pravděpodobnostních distribucí) budeme využívat. V podsekci 1.1.3 již bylo zmíněno, že oborová konvence nevyžaduje značit náhodné veličiny tak, jak jsme z teorie pravděpodobnosti zvyklí. Stejně tak tomu bude i u jejich odpovídajích hustot pravděpodobností. Díky tomuto faktu a výkladu podsekce 1.1.5 plyne tvrzení:

$$\boxed{\mathbf{X} \sim \mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \Leftrightarrow p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \Leftrightarrow p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}_d(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})}$$

1.2 Optimalizace

Zde popíšeme základní numerické metody a nástroje, které budeme v rámci optimalizace používat. Cílem optimalizace je najít, respektive naučit se, optimální parametry jisté funkce tak, aby její hodnota v závislosti na těchto parametrech byla vždy co nejmenší (1.27). Takové funkci budeme říkat **ztrátová funkce** (*loss function*) a značit $L(\boldsymbol{\theta})$. Optimální parametry lze nalézt jako argument jejího minima:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}). \tag{1.27}$$

1.2.1 Metoda nejstrmějšího sestupu

Metoda nejstrmějšího sestupu (též *Gradient Descent*) je iterační optimalizační metoda prvního řádu k nalezení lokálního minima diferencovatelné funkce m proměnných, kde $m \in \mathbf{N}$ [1].

Nechť je nyní vektor parametrů $\boldsymbol{\theta}$ označen jako vektor proměnných a $m = d+1$, tedy $\boldsymbol{\theta} = (w_0, \ldots, w_d)^T$. Cílem této metody je nalézt takové $\hat{\boldsymbol{\theta}}$, pro které $L(\boldsymbol{\theta})$ nabývá minimální hodnoty. Jinými slovy, chceme splnit vztah:

$$\begin{gathered}\left(\frac{\partial L(\boldsymbol{\theta})}{\partial w_0}, \frac{\partial L(\boldsymbol{\theta})}{\partial w_1}, \ldots, \frac{\partial L(\boldsymbol{\theta})}{\partial w_d}\right) = (0, 0, \ldots, 0) \\ \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \mathbf{0}.\end{gathered} \tag{1.28}$$

Mějme tedy nějaký bod $\boldsymbol{\theta}_0$ funkce $L(\boldsymbol{\theta})$ jako výchozí. Každá další poloha bodu $\boldsymbol{\theta}_0$ ve směru záporného gradientu funkce se spočítá pomocí předchozí iterace na základě vztahu

$$\boldsymbol{\theta}_0^{(\tau+1)} = \boldsymbol{\theta}_0^{(\tau)} - \eta \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_0^{(\tau)}), \tag{1.29}$$

kde τ je iterační číslo a η představuje učící paratmetr (tzv. krok nebo-li *learning rate*). V klasické verzi této metody je učící parametr ve formě *skaláru*. Jelikož je $L(\boldsymbol{\theta})$ hladká spojitá funkce, pak bod, pro

který platí rovnice $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_0) = \mathbf{0}$ a je tím pádem splněna iterační rovnost $\boldsymbol{\theta}_0^{(\tau+1)} = \boldsymbol{\theta}_0^{(\tau)}$, označíme jako podežrelý z extrému. Je to nutná, ale ne postačující podmínka. Body, kde gradient vymizí, se nazývají stacionární a mohou být dále klasifikovány jako minima, maxima, nebo sedlové body. Pokud se nám povede doiterovat do minima funkce, je bod $\boldsymbol{\theta}_0$ označen jako $\hat{\boldsymbol{\theta}}$ a platí (1.27).

Poznámka. V této práci bude vektor $\boldsymbol{\theta}$ reprezentovat vektor parametrů (*regrese*) nebo vektor vah (*neuronové sítě*).

Klasická verze nejstrmějšího sestupu není tak efektivní a rychlá, jak bychom v praxi potřebovali. Zejména kvůli parametru η, který je v tomto případě skalárem. Je opravdu složité a téměř nemožné popsat jedním, neadaptivním číslem všechny dimenze. Proto byly zavedeny pokročilejší, adaptivní metody využívající vektorový učící parametr měnící se každou iterací, jehož složky jsou adaptivní v rámci každé dimenze, tj. $\boldsymbol{\eta}_{(\tau)} = (\eta_{(\tau)_0}, \eta_{(\tau)_1}, \ldots, \eta_{(\tau)_d})^T$.

Stochastický nejstrmější sestup

Uvažujme nyní funkci $L(\boldsymbol{\theta})$, na kterou klademe požadavek (1.30)

$$L(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L_i(\boldsymbol{\theta}), \tag{1.30}$$

kde L_i se typicky pojí s i–tým pozorováním v data setu. Vyberme náhodně nějaký index j z množiny indexů $\{1, 2, \ldots, n\}$ s pravděpodobností $p(j = i) = \frac{1}{n}$ a spočtěme střední hodnotu jeho obecného gradientu:

$$\mathsf{E}_{p(j=i)} \left[\nabla_{\boldsymbol{\theta}} L_j(\boldsymbol{\theta}) \right] = \sum_{i=1}^{n} p(j = i) \nabla_{\boldsymbol{\theta}} L_i(\boldsymbol{\theta}) = \sum_{i=1}^{n} \frac{1}{n} \nabla_{\boldsymbol{\theta}} L_i(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} L_i(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}). \tag{1.31}$$

V (1.31) lze vidět, že při jakémkoli výběru indexu j z množiny indexů vždy dostaneme **nestranný odhad** obecného gradientu. Tímto jsme získali **stochastický gradient**, díky čemuž můžeme po přidání adaptivního $\boldsymbol{\eta}_{(\tau)}$ upravit (1.29) do tvaru

$$\boldsymbol{\theta}_0^{(\tau+1)} = \boldsymbol{\theta}_0^{(\tau)} - \boldsymbol{\eta}_{(\tau)} \nabla_{\boldsymbol{\theta}} L_j(\boldsymbol{\theta}_0^{(\tau)}), \tag{1.32}$$

kde počet iterací τ je nezávislý na velikosti n. Této úpravě říkáme **stochastický nejstrmější sestup** (*Stochastic Gradient Descent* [17]). K zajištění konvergence metody je třeba, aby $\boldsymbol{\eta}_{(\tau)} \to \mathbf{0}$.

ADAM

ADAM (*Adaptive Moment Estimation*) je vylešení klasické metody nejstrmějšího sestupu o adaptivní vektorové $\boldsymbol{\eta}_{(\tau)}$ a druhý moment gradientu, což v určitých případech podstatně zrychluje učení. Proto budeme algoritmus ADAM hojně využívat hlavně v praktických příkladech.

1.2.2 Metoda nejmenších čtverců

Metoda nejmenších čtverců (*MNČ*) je matematicko–stochastická metoda pro aproximaci řešení předurčených soustav rovnic. Je ekvivalentní lineární regresi.

Nejjednodušší model pro regresi, který obsahuje lineární kombinaci proměnných, je

$$y(\mathbf{x}, \boldsymbol{\theta}) = w_0 + w_1 x_1 + \ldots + w_d x_d, \tag{1.33}$$

kde $\boldsymbol{\theta} = (w_0, w_1, \ldots, w_d)^T$ je vektor parametrů. Lze zapsat jako $y(\mathbf{x}, \boldsymbol{\theta}) = w_0 + \sum_{j=1}^{d} w_j x_j$.

Ovšem pouze s tímto si, bohužel, nevystačíme. Mějme tedy n–prvkovou množinu pozorování X, tedy $(\mathbf{x}_i, y_i(\mathbf{x}_i))$, kde $i \in \hat{n}$. To znamená, že nyní máme soustavu n rovnic. Vektor proměnných i–tého pozorování můžeme zapsat jako $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^T$. Zapišme nyní předpis pro i–té pozorování závisející na možných $d + 1$ parametrech:

$$y_i(\mathbf{x}_i, \boldsymbol{\theta}) = w_0 + w_1 x_{i1} + \ldots + w_d x_{id}, \forall i \in \hat{n} \tag{1.34}$$

Rozšíříme-li tento systém uvažováním lineární kombinace fixních nelineárních funkcí, pak ho lze zapsat pomocí sumy:

$$y_i(\mathbf{x}_i, \boldsymbol{\theta}) = w_0 + \sum_{j=1}^{d} w_j \phi_j(\mathbf{x}_i), \forall i \in \hat{n}, \tag{1.35}$$

kde $\phi_j(\mathbf{x}_i)$ představují tzv. *bázové funkce*. Zadefinováním $\phi_0(\mathbf{x}_i) := 1 \ \forall i \in \hat{n}$ můžeme sumu upravit do tvaru:

$$y_i(\mathbf{x}_i, \boldsymbol{\theta}) = \sum_{j=0}^{d} w_j \phi_j(\mathbf{x}_i) = \boldsymbol{\theta}^T \Phi(\mathbf{x}_i),$$

kde $\Phi(\mathbf{x}_i) = (\phi_0(\mathbf{x}_i), \ldots, \phi_d(\mathbf{x}_i))$ představuje vektor bázových funkcí pro i–té pozorování.

Poznámka. Povšimněme si, že index j, jenž označuje určitou složku ve vektoru parametrů $\boldsymbol{\theta}$, probíhá od 0 až po nějaké $d \in \mathbf{N}$. To implikuje tvrzení, že parametrů je právě $d + 1$. Na tyto parametry je kladena podmínka, aby jejich počet v modelu byl menší, nebo roven n.

Klasické pojetí MNČ

Vezměme nyní v úvahu polynomické bázové funkce $\phi_j(\mathbf{x}_i) = \left(\mathbf{x}_{ij}\right)^j$. Nechť je dáno n pozorování, $n \in \mathbf{N}$ a $\mathbf{y} = (y_1, \ldots, y_n)^T \in \mathbf{R}^n$. Nechť dále $d \in \mathbf{N}_0$ je stupeň prokládaného polynomu (v případě fitování dat), $\boldsymbol{\theta} = (w_0, w_1, w_2, \ldots, w_d)^T \in \mathbf{R}^{d+1}$ je vektor parametrů, $(x_{i1}, x_{i2}, \ldots, x_{id})^T$ je vektor d proměnných i–tého pozorování a matice $\mathbb{X} = (\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_n))^T \in \mathbf{R}^{n \times d+1}$ je tzv. matice bázových funkcí:

$$\begin{aligned}
y_1 &= w_0\phi_0(\mathbf{x}_1) + w_1\phi_1(\mathbf{x}_1) + w_2\phi_2(\mathbf{x}_1) + w_3\phi_3(\mathbf{x}_1) + w_4\phi_4(\mathbf{x}_1) + \ldots + w_d\phi_d(\mathbf{x}_1) \\
y_2 &= w_0\phi_0(\mathbf{x}_2) + w_1\phi_1(\mathbf{x}_2) + w_2\phi_2(\mathbf{x}_2) + w_3\phi_3(\mathbf{x}_2) + w_4\phi_4(\mathbf{x}_2) + \ldots + w_d\phi_d(\mathbf{x}_2) \\
&\vdots \\
y_n &= w_0\phi_0(\mathbf{x}_n) + w_1\phi_1(\mathbf{x}_n) + w_2\phi_2(\mathbf{x}_n) + w_3\phi_3(\mathbf{x}_n) + w_4\phi_4(\mathbf{x}_n) + \ldots + w_d\phi_d(\mathbf{x}_n).
\end{aligned}$$

Po dosazení polynomických bázových funkcí dostaneme soustavu:

$$\begin{aligned}
y_1 &= w_0\,(x_{10})^0 + w_1\,(x_{11})^1 + w_2\,(x_{12})^2 + w_3\,(x_{13})^3 + w_4\,(x_{14})^4 + \ldots + w_d\,(x_{1d})^d \\
y_2 &= w_0\,(x_{20})^0 + w_1\,(x_{21})^1 + w_2\,(x_{22})^2 + w_3\,(x_{23})^3 + w_4\,(x_{24})^4 + \ldots + w_d\,(x_{2d})^d \\
&\vdots \\
y_n &= w_0\,(x_{n0})^0 + w_1\,(x_{n1})^1 + w_2\,(x_{n2})^2 + w_3\,(x_{n3})^3 + w_4\,(x_{n4})^4 + \ldots + w_d\,(x_{nd})^d\,.
\end{aligned}$$

Což lze zapsat maticovým zápisem:

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} 1 & (x_{11})^1 & (x_{12})^2 & \ldots & (x_{1d})^d \\ 1 & (x_{21})^1 & (x_{22})^2 & \ldots & (x_{2d})^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (x_{n1})^1 & (x_{n2})^2 & \ldots & (x_{nd})^d \end{pmatrix}}_{\mathbb{X}} \cdot \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{pmatrix}}_{\boldsymbol{\theta}}.$$

Odhad vektoru $\boldsymbol{\theta}$

Ve skutečnosti ale řešíme soustavu $\mathbf{y} = \mathbb{X}\boldsymbol{\theta} + \mathbf{e}$, kde $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ je tzv. *residuum*, které se snažíme minimalizovat. Z euklidovské normy plyne: $\|\mathbf{e}\|_2 = \sqrt{\sum_{j=1}^{n} e_i^2}$. Nyní normu minimalizujme:

$$\arg\min \|\mathbf{e}\|_2 = \arg\min \|\mathbf{e}\|_2^2 = \arg\min \sum_{i=1}^{n} e_i^2 = \arg\min \left(\mathbf{e}^T \mathbf{e}\right).$$

Dosadíme do výrazu a zderivujeme podle parametru $\boldsymbol{\theta}$. Nesmíme zapomenout, že výraz derivujeme podle vektoru. Pro přímočarost si označíme součin $\mathbf{e}^T\mathbf{e}$ jako f.

$$\begin{aligned}
\frac{\partial f}{\partial \boldsymbol{\theta}} &= \frac{\partial \left(\mathbf{y} - \mathbb{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbb{X}\boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}} = \frac{\partial \left(\mathbf{y}^T\mathbf{y} - \boldsymbol{\theta}^T\mathbb{X}^T\mathbf{y} - \mathbf{y}^T\mathbb{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbb{X}^T\mathbb{X}\boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}} = \\
&= -\mathbb{X}^T\mathbf{y} - \mathbb{X}^T\mathbf{y} + 2\mathbb{X}^T\mathbb{X}\boldsymbol{\theta} = -2\mathbb{X}^T\mathbf{y} + 2\mathbb{X}^T\mathbb{X}\boldsymbol{\theta} = -2\mathbb{X}^T\mathbf{y} + 2\mathbb{X}^T\mathbb{X}\boldsymbol{\theta}
\end{aligned}$$

Nyní výraz položme rovno nulovému vektoru a upravme:

$$\begin{aligned}
-2\mathbb{X}^T\mathbf{y} + 2\mathbb{X}^T\mathbb{X}\boldsymbol{\theta} &\stackrel{!}{=} \mathbf{0} \\
\mathbb{X}^T\mathbb{X}\boldsymbol{\theta} &= \mathbb{X}^T\mathbf{y} \\
\hat{\boldsymbol{\theta}} &= (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbf{y}.
\end{aligned} \tag{1.36}$$

Tímto způsobem jsme nalezli maximální věrohodný odhad parametru $\boldsymbol{\theta}$ (označme $\hat{\boldsymbol{\theta}}$), jenž je díky předpokladu lineární nezávislosti sloupců jednoznačným řešením.

Bayesovské pojetí MNČ

Uvažujme soustavu $\mathbf{y} = \mathbb{X}\boldsymbol{\theta}$, ke které nyní přidáme vektor $\mathbf{e} \in \mathbf{R}^n$ reprezentující šum řídící se rozdělením $\mathbf{e} \sim \mathcal{N}_n(\mathbf{0}, \mathbb{I})$ (viz.1.1.5). Vektoru $\mathbf{y}$ nyní budeme říkat vektor dat. Po jednoduché úpravě dostáváme:

$$\mathbf{e} = \mathbf{y} - \mathbb{X}\boldsymbol{\theta}. \tag{1.37}$$

Složky náhodného vektoru $\mathbf{e}$ jsou nezávislé, tzn. $p(\mathbf{e}) = \prod_{i=1}^{n} p(e_i)$. Jistě tedy můžeme psát:

$$p(\mathbf{e}) \overset{1.26}{\propto} \exp\left(-\frac{1}{2}\mathbf{e}^T\mathbf{e}\right) = \exp(-\frac{1}{2}\sum_{i=1}^{n} e_i^2) = \prod_{i=1}^{n} \exp(-\frac{1}{2}e_i^2). \tag{1.38}$$

Rádi bychom teď podobně jako v podsekci 1.2.2 co možná nejvěrohodněji odhadli parametr $\boldsymbol{\theta}$. Jelikož do hry vstoupila pravděpodobnostní rozdělení, je třeba k problému přistoupit *bayesovským přístupem*. Výstupem nám tedy bude pravděpodobnostní rozdělení vektoru $\boldsymbol{\theta}$, nikoli vektor samotný jako tomu bylo u klasické MNČ. Díky tomu můžeme kvantifikovat naši míru neurčitosti ohledně modelu.

Rádi bychom maximalizovali pravděpodobnost odhadu $\boldsymbol{\theta}$ za předpokladu, že máme data $\mathbf{y}$ a matici bázových funkcí, což nám dává vztah:

$$p(\boldsymbol{\theta}|\mathbf{y}, \mathbb{X}) \overset{1.5}{=} \frac{p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X})p(\boldsymbol{\theta}|\mathbb{X})}{p(\mathbf{y}|\mathbb{X})} \overset{1.7}{\propto} p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X})p(\boldsymbol{\theta}|\mathbb{X}). \tag{1.39}$$

Dosazením (1.37) do (1.38) získáme pravděpodobnostní rozdělení prvního členu součinu:

$$p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X}) \propto \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbb{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbb{X}\boldsymbol{\theta})\right). \tag{1.40}$$

Abychom mohli počítat dále, je třeba určit druhý člen součinu. Zpravidla můžeme zvolit, jakým rozdělením se bude řídit $p(\boldsymbol{\theta}|\mathbb{X}) \equiv p(\boldsymbol{\theta})$. Rádi bychom ovšem vnesli do modelu novou informaci, kterou bychom mohli korigovat parametrem α (bude vysvětleno a použito později). Zvolme $\boldsymbol{\theta} \sim \mathcal{N}_{d+1}(\mathbf{0}, \alpha^{-1}\mathbb{I})$:

$$p(\boldsymbol{\theta}) \propto \exp(-\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}\alpha). \tag{1.41}$$

Dosaďme nyní (1.40) a (1.41) do rovnice (1.39), tzn.

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X})p(\boldsymbol{\theta}|\mathbb{X}) &\propto \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbb{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbb{X}\boldsymbol{\theta})\right)\exp\left(-\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}\alpha\right) \propto \\ &\propto \exp\left(-\frac{1}{2}(\mathbf{y}^T\mathbf{y} - \boldsymbol{\theta}^T\mathbb{X}^T\mathbf{y} - \mathbf{y}^T\mathbb{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbb{X}^T\mathbb{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\boldsymbol{\theta}\alpha)\right) \propto \\ &\propto \exp\left(-\frac{1}{2}(\mathbf{y}^T\mathbf{y} - \boldsymbol{\theta}^T\mathbb{X}\mathbf{y} - \mathbf{y}^T\mathbb{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T(\mathbb{X}^T\mathbb{X} + \alpha\mathbb{I})\boldsymbol{\theta})\right). \end{aligned} \tag{1.42}$$

Je nutné odhadnout, čemu se budou jednotlivé členy argumentu exp v (1.42) rovnat. Předpokládejme, že výsledná $p(\boldsymbol{\theta}|\mathbf{y}, \mathbb{X})$ má pravděpodobnostní rozdělení $\mathcal{N}(\hat{\boldsymbol{\theta}}, \Sigma)$, a že kvadratická forma lze upravit do tohoto tvaru, tzn.

$$\begin{aligned} p(\boldsymbol{\theta}|\mathbf{y}, \mathbb{X}) &\propto \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + z\right) \propto \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right)\exp(z) \\ &\propto \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right), \end{aligned} \tag{1.43}$$

kde z představuje možný zbytek po úpravě v rámci kvadratických forem. Ovšem jeho příspěvek nás nezajímá a můžeme ho zanedbat.

Roznásobením (1.43) dostaneme:

$$\exp\left(-\frac{1}{2}(\boldsymbol{\theta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta} - \boldsymbol{\theta}^T\boldsymbol{\Sigma}^{-1}\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\hat{\boldsymbol{\theta}})\right). \tag{1.44}$$

Úpravou a porovnáním vztahů (1.42) a (1.44) lze odvodit, že $\boldsymbol{\Sigma}^{-1} = \mathbb{X}^T\mathbb{X} + \alpha\mathbb{I}$. Použijeme též fakt, že kovarianční matice je symetrická, tudíž $\boldsymbol{\Sigma}^T = \boldsymbol{\Sigma}$. Postup výpočtu $\hat{\boldsymbol{\theta}}$ je uveden v (1.45):

$$\begin{aligned}
-\mathbf{y}^T\mathbb{X}\boldsymbol{\theta} &= -\hat{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta} \\
\mathbf{y}^T\mathbb{X} &= \hat{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}^{-1} \\
\left(\mathbf{y}^T\mathbb{X}\boldsymbol{\Sigma}\right)^T &= \hat{\boldsymbol{\theta}} \\
\hat{\boldsymbol{\theta}} &= \boldsymbol{\Sigma}\mathbb{X}^T\mathbf{y} \\
\hat{\boldsymbol{\theta}} &= (\mathbb{X}^T\mathbb{X} + \alpha\mathbb{I})^{-1}\mathbb{X}^T\mathbf{y}.
\end{aligned} \tag{1.45}$$

Dosazením vypočtených hodnot $\hat{\boldsymbol{\theta}}$ a $\boldsymbol{\Sigma}$ do (1.43), dostaneme aposteriorní pravděpodobnostní rozdělení (tzv. *posterior distribution*) pro parametry:

$$p(\boldsymbol{\theta}|\mathbf{y}, \mathbb{X}) \propto \mathcal{N}\left(\boldsymbol{\theta}|(\mathbb{X}^T\mathbb{X} + \alpha\mathbb{I})^{-1}\mathbb{X}^T\mathbf{y}, (\mathbb{X}^T\mathbb{X} + \alpha\mathbb{I})^{-1}\right). \tag{1.46}$$

1.2.3 Elbo

Evidence lower bound (též Elbo) patří mezi variační bayesovské metody strojového učení [23]. Úzce souvisí s pojmem *KL divergence* popsaným v 1.1.4. Díky této optimalizační metodě budeme schopni minimalizovat míru odlišnosti mezi přesnou a aproximativní pravděpodobnostní distribucí.

Zkusme nyní rozvést pojmy, které již známe. Nechť je dán set n nezávislých, stejně rozdělených dat a množina pozorování $X = \{\mathbf{x}_i \mid i \in \hat{n}\}$. Naším úkolem bude najít aproximativní pravděpodobnostní distribuci pro *aposteriorní distribuci* $p(\mathbf{z}|\mathbf{x})$ a $p(\mathbf{x})$. Zlogaritmujme a upravme:

$$\begin{aligned}
\ln p(\mathbf{x}) &\overset{1,12}{=} \int q(\mathbf{z}) \ln\left(p(\mathbf{x})\right) d\mathbf{z} \overset{1,2}{=} \int q(\mathbf{z}) \ln\left(\frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}\right) d\mathbf{z} = \int q(\mathbf{z}) \ln\left(\frac{p(\mathbf{x},\mathbf{z})q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})q(\mathbf{z})}\right) d\mathbf{z} = \\
&= \int q(\mathbf{z}) \left(\ln\left\{\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right\} - \ln\left\{\frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}\right\}\right) d\mathbf{z} = \\
&= \int q(\mathbf{z}) \left(\ln\left\{\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right\}\right) d\mathbf{z} - \int q(\mathbf{z}) \left(\ln\left\{\frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}\right\}\right) d\mathbf{z} = \\
&= \mathcal{L}(q(\mathbf{z})) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})).
\end{aligned} \tag{1.47}$$

Výrazu $\mathcal{L}(q)$ se přezdívá *lower bound* (dolní mez) a tvoří základ naší metody. Ze znalosti vlastností (1.22) víme, že hodnota *KL divergence* je nezáporná. Díky maximalizaci $\mathcal{L}(q)$ minimalizujeme *KL divergenci* mezi danými distribucemi a naopak. Jelikož je $\ln p(\mathbf{x})$ vzhledem k itegraci konstanta, musí platit, že pokud jeden z výrazů roste, druhý musí automaticky klesat. Pokud dovolíme jakoukoli volbu $q(\mathbf{z})$, pak maximum dolní meze nastává, když *KL divergence* zmizí, což je možné pouze, pokud se *aposteriorní distribuce* $p(\mathbf{z}|\mathbf{x})$ rovná $q(\mathbf{z})$.

1.2.4 Bayesovská predikce dat

Nechť $\mathbf{y}$ je n–dimenzionální vektor dat, tedy $\mathbf{y} = (y_1, \ldots, y_n)^T$, a jemu odpovídajících n pozorování $X = \{\mathbf{x}_i \mid i \in \hat{n}\}$. Řekněme, že známe hodnotu $\mathbf{x}_{n+1}$. Snažme se nyní nalézt *prediktivní* pravděpodobnostní distribuci neznámé cílové hodnoty $y_{n+1} \overset{ozn.}{=} y^+$. Označme $X' = \{\mathbf{x}_i \mid i \in \widehat{n+1}\}$ jako množinu $n+1$ pozorování. Chtěli bychom predikovat nové cílové hodnoty na základě výše zmíněných znalostí.

Nechť opět $\boldsymbol{\theta} \sim \mathcal{N}_{d+1}(\hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma})$, tedy $p(\boldsymbol{\theta}|\mathbf{y}, X) = \mathcal{N}(\hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma})$. Pak nové cílové hodnoty lze spočítat marginalizací přes proměnné, kterými si nejsme jisti, tedy přes parametry $\boldsymbol{\theta}$, a posléze užitím součinového pravidla:

$$p(y^+|X', \mathbf{y}) = \int p(y^+, \boldsymbol{\theta}|X', \mathbf{y}) d\boldsymbol{\theta} \overset{1.2}{=} \int p(y^+|\boldsymbol{\theta}, \mathbf{x}_{n+1}) p(\boldsymbol{\theta}|X, \mathbf{y}) d\boldsymbol{\theta}. \tag{1.48}$$

Pokud nastane případ, že výsledný integrál v (1.48) nebude analyticky řešitelný, lze využít vhodné aproximace pomocí *diracovských* pulzů (1.49) a výpočtem pomocí *Monte Carlo* metody (1.50), tj.

$$p(\boldsymbol{\theta}|X, \mathbf{y}) \approx \frac{1}{n} \sum_{i=1}^{n} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}), \tag{1.49}$$

$$\begin{aligned} p(y^+|X', \mathbf{y}) = \int p(y^+|\boldsymbol{\theta}, \mathbf{x}_{n+1}) p(\boldsymbol{\theta}|X, \mathbf{y}) d\boldsymbol{\theta} \overset{1.49}{\approx} \int p(y^+|\boldsymbol{\theta}, \mathbf{x}_{n+1}) \frac{1}{n} \sum_{i=1}^{n} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}) d\boldsymbol{\theta} \approx \\ \approx \frac{1}{n} \sum_{i=1}^{n} \int p(y^+|\boldsymbol{\theta}, \mathbf{x}_{n+1}) \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}) d\boldsymbol{\theta} \approx \frac{1}{n} \sum_{i=1}^{n} p(y^+|\boldsymbol{\theta}^{(i)}, \mathbf{x}_{n+1}). \end{aligned} \tag{1.50}$$

V (1.50) lze vidět, že díky znalosti hodnoty $\mathbf{x}_{n+1}$ a apriorního rozdělení parametrů dokážeme *predikovat* pravděpodobnostní distribuci následující cílové hodnoty jako průměr přes všechny známé pravděpodobnostní distribuce.

1.3 Agregace

Již víme, že každé složce vektoru $\mathbf{y}$ je přiřazen právě jeden prvek z množiny X, tedy:

$$\begin{aligned} \mathbf{x}_1 &\mapsto y_1 \\ \mathbf{x}_2 &\mapsto y_2 \\ &\vdots \\ \mathbf{x}_n &\mapsto y_n \\ X &\mapsto \mathbf{y}. \end{aligned}$$

V případě, kdy máme přesně definované, jakým způsobem prvky X ke složkám $\mathbf{y}$ přiřazujeme, problém nenastává. Co když je ale potřeba snížit počet dimenzí vektoru $\mathbf{y}$ na výstupu?

$$X \overset{?}{\mapsto} \mathbf{y'}, \quad \text{kde } \mathbf{y'} \in \mathbf{R}^k, \quad k < n \quad (k \text{ je fixní}).$$

Tuto operaci nazveme **agregací**, jež je definovaná pomocí **agregační funkce** (nebo-li agregačního operátoru). Mezi agregační operátory patří například softmax, klasické maximum, minimum nebo aritmetický či vážený průměr. Jak ale najít optimální agregační operátor nevíme. Obrovská výhoda této operace je zajištění fixního počtu dimenzí na výstupu. Taková situace často nastává například ve vrstvení neuronových sítí, či na různých úrovních stromových struktur [1].

1.4 Neuronové sítě

Již jsme se zabývali standardní regresí s polynomickými bázovými funkcemi, vektorově zapsáno jako:

$$\mathbf{y} = \mathbb{X}\boldsymbol{\theta}. \tag{1.51}$$

Nechť $\mathbf{f}(\mathbf{x})$ je nyní nějaká vektorová transformační funkce. Vložme tuto funkci do (1.51), čímž získáme

$$\mathbf{y} = \mathbf{f}\left(\mathbb{X}\boldsymbol{\theta}\right). \tag{1.52}$$

Rozepsáním vztahu (1.52) dostaneme sadu n transformačních rovnic, tj.

$$\begin{aligned}
y_1(\mathbf{x}_1, \boldsymbol{\theta}) &= f_1\Big(w_0\phi_0(\mathbf{x}_1) + w_1\phi_1(\mathbf{x}_1) + w_2\phi_2(\mathbf{x}_1) + w_3\phi_3(\mathbf{x}_1) + w_4\phi_4(\mathbf{x}_1) + \ldots + w_d\phi_d(\mathbf{x}_1)\Big) \\
y_2(\mathbf{x}_2, \boldsymbol{\theta}) &= f_2\Big(w_0\phi_0(\mathbf{x}_2) + w_1\phi_1(\mathbf{x}_2) + w_2\phi_2(\mathbf{x}_2) + w_3\phi_3(\mathbf{x}_2) + w_4\phi_4(\mathbf{x}_2) + \ldots + w_d\phi_d(\mathbf{x}_2)\Big) \\
&\vdots \\
y_n(\mathbf{x}_n, \boldsymbol{\theta}) &= f_n\Big(w_0\phi_0(\mathbf{x}_n) + w_1\phi_1(\mathbf{x}_n) + w_2\phi_2(\mathbf{x}_n) + w_3\phi_3(\mathbf{x}_n) + w_4\phi_4(\mathbf{x}_n) + \ldots + w_d\phi_d(\mathbf{x}_n)\Big),
\end{aligned}$$

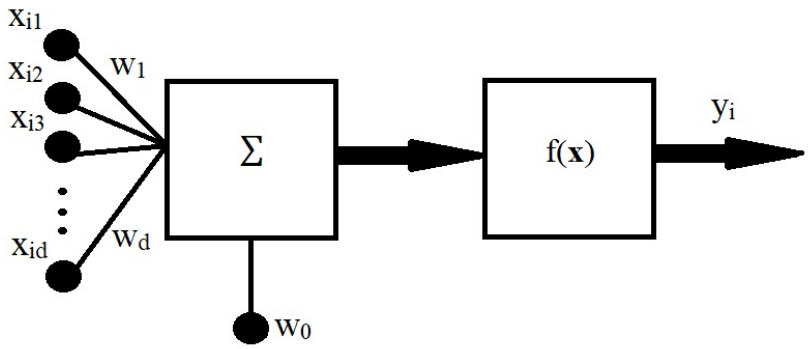
což můžeme pro přehlednost napsat jako

$$y_i(\mathbf{x}_i, \boldsymbol{\theta}) = f_i\left(\sum_{j=0}^{d} w_j\phi_j(\mathbf{x}_i)\right), \forall i \in \hat{n}. \tag{1.53}$$

V (1.53) jsme získali matematický popis jednoho **neuronu**, jenž je graficky znázorněn na Obrázku 1.4. Jedná se vlastně o zobrazení $\mathbf{R}^d \mapsto \mathbf{R}$. Pokud rozepíšeme argument uvnitř transformační funkce, tj.

$$y_i(\mathbf{x}_i, \boldsymbol{\theta}) = f_i\left(\sum_{j=1}^{d} w_j\phi_j(\mathbf{x}_i) + w_0\right), \tag{1.54}$$

můžeme si povšimnout konstanty w_0, které se jinak říká **práh** (*bias*). Díky němu je možné neuron aktivovat. Ostatní parametry nazýváme **vahami** (*weights*).



Obrázek 1.4: Jednoduché schéma umělého neuronu s použitím identických bázových funkcí.

Pokud bychom po transformaci požadovali vektorový výstup, hovoříme o **neuronové síti**. V takovém případě požadujeme, aby vektorová transformační funkce měla shodné všechny její složky, tzn. $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x}))^T = (f(\mathbf{x}), \ldots, f(\mathbf{x}))^T$. Složky transformační vektorové funkce $\mathbf{f}(.)$ nazýváme **aktivační funkcí**. Sítě můžeme dále vrstvit a zvyšovat nároky na jejich učení.

Transformací v (1.52) jsme tedy vytvořili **jednovrstvou neuronovou síť** s aktivační funkcí $f(\mathbf{x})$.

Poznámka. Povšimněme si, že pokud za $\mathbf{f}(\mathbf{x})$ v (1.52) zvolíme identickou vektorovou funkci [10] dostaneme známý vztah $\mathbf{y} = \mathbb{X}\boldsymbol{\theta}$. Můžeme tedy říct, že lineární regrese je ekvivalentní pojmenování *jednovrstvé neuronové sítě s identickou aktivační funkcí*.

1.4.1 Aktivační funkce

V případě jednoho neuronu je možné si tuto funkci představit jako klasickou funkci více proměnných, kterou známe z matematické analýzy. Jejími argumenty jsou **vstupy neuronu** $x_1, \ldots, x_d$ vážené příslušnými **vahami** $w_1, \ldots, w_d$ doplněné o **prah** w_0.

Pokud budujeme neuronovou síť, matematicky se aktivační funkce změní na vektorovou funkci, díky které dokážeme na výstupu sítě dostat vektor dat. Tento výstup může být vstupem do další vrstvy, čímž se sítě vrství.

Uveďme pro ukázku některé typy používaných aktivačních funkcí (zjednodušeně jako funkce jednoho vstupu):

- **Identita** – používaná v lineární regresi

$$f_{\mathrm{Id}}(x) := x \tag{1.55}$$

- **Sigmoida** (též zvaná jako logistická funkce) – používaná především v logistické regresi

$$f_{\sigma}(x) := \frac{1}{1 + \exp(-x)} \tag{1.56}$$

- **Hyperbolický tangens**

$$f_{\tanh}(x) := \tanh(x) \tag{1.57}$$

- **Jednotkový skok** – definován pomocí Heavisideovy[11] funkce

$$f_{\Theta}(x) := \Theta(x) = \begin{cases} 1 & \text{pro} \quad x > 0 \\ 0 & \text{pro} \quad x \leq 0 \end{cases} \tag{1.58}$$

- **ReLU** (*Rectified Linear Unit*) – hojně využívaná, ovšem nediferencovatelná v 0

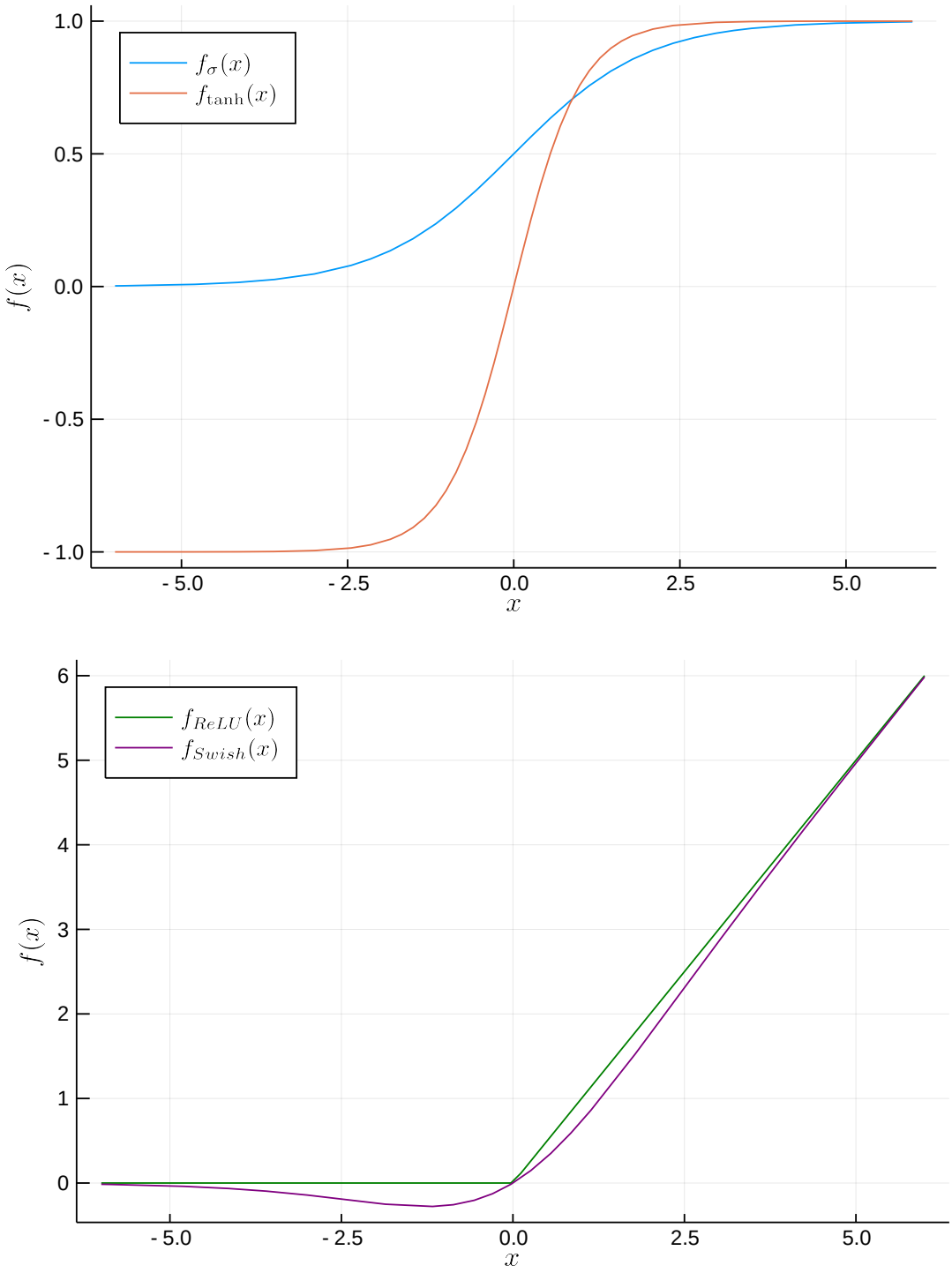
$$f_{\mathrm{ReLU}}(x) := \max(0, x) \tag{1.59}$$

- **Swish** – spojitá, nemonotonní aproximace ReLU. Je definována jako:

$$f_{\mathrm{Swish}}(x) := x \cdot \frac{1}{1 + \exp(-x)} \tag{1.60}$$

[10] značenou též někdy jako **Id**

[11] *Oliver Heaviside* (1850–1925)



Obrázek 1.5: Grafické znázornění některých výše zmíněných aktivačních funkcí.

1.5 Teorie grafů

Abychom mohli stavět na složitějších typech datových struktur, je třeba náhlednout do jednoho z oborů diskrétní matematiky, jenž poskytuje silný nástroj, jak efektivně popsat a řešit zadané problémy. Prvně řádně zadefinujme některé důležité definice. Základní definice byly čerpány z [5, 20] a upraveny do konzistentního značení.

Definice 1.5.1. (Graf) **Grafem** rozumíme uspořádanou dvojici množiny vrcholů V a množiny hran E, tj. $G = (V, E)$. Každá hrana je povinně určena dvěma vrcholy a volitelně směrem nebo vahou.

Poznámka. Velikost množiny V nazveme také *počtem vrcholů*, značíme $|V|$, a velikost množiny E *počtem hran*, $|E|$. Vrcholy zpravidla znázorňujeme jako kroužky a hrany jako úsečky.

Definice 1.5.2. (Orientovaný graf) Pokud jsou hrany grafu doplněny o tzv. **šipky**, které určují jejich směr, hovoříme o **orientovaném grafu**.

Poznámka. Hrana se šipkami na obou koncích značí směr z V_i do V_j a naopak. V počtu hran se každá taková hrana počítá za dvě.

Definice 1.5.3. (Cesta) **Cestou v grafu** G nazýváme posloupnost vrcholů a hran $(V_0, E_1, V_1, \ldots, E_t, V_t)$, kde vrcholy $V_0, \ldots, V_t$ jsou navzájem různé vrcholy G a $\forall i \in \hat{t}$ platí $E_i = \{V_{i-1}, V_i\} \in E$.

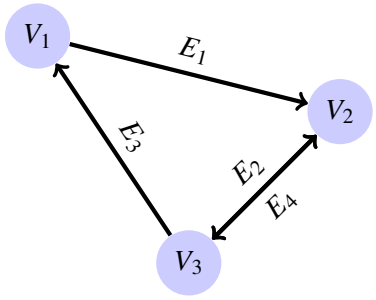
Definice 1.5.4. (Souvislost grafu) Řekneme, že graf G je **souvislý**, pokud pro každé jeho dva vrcholy existuje v G mezi těmito vrcholy cesta.

Definice 1.5.5. (Kružnice v grafu) **Kružnicí (cyklem) v grafu** G nazýváme posloupnost vrcholů a hran $(V_0, E_1, V_1, \ldots, E_t, V_t = V_0)$, kde vrcholy $V_0, \ldots, V_{t-1}$ jsou navzájem různé vrcholy G a $\forall i \in \hat{t}$ platí $E_i = \{V_{i-1}, V_i\} \in E$.

Definice 1.5.6. (Ohodnocení hran) Množinovou funkci $\mathrm{W} : E(G) \to \mathbf{R}$ nazveme **ohodnocením hran**.

Definice 1.5.7. (Vzdálenost v grafu) Nechť G je graf. Pro vrcholy V_i a V_j definujeme číslo $d_G(V_i, V_j)$ jako nejkratší délku cesty z V_i do V_j. Toto číslo nazveme **vzdáleností vrcholů** V_i a V_j v G.

Poznámka. Funkce $d_G : V \times V \to \mathbf{R}_0^+$ obecně splňuje axiomy **metriky** a nazýváme ji *metrika grafu* G.



Obrázek 1.6: Ukázka grafu se třemi vrcholy a čtyřmi hranami

1.5.1 Stromy

Po stručném úvodu do teorie grafů můžeme přejít k více konkrétnímu příkladu grafu, a to *stromu*. Jelikož je tedy *strom* pouze konkrétní realizace grafu, vztahují se na něj všechny definice vyslovené v sekci 1.5.

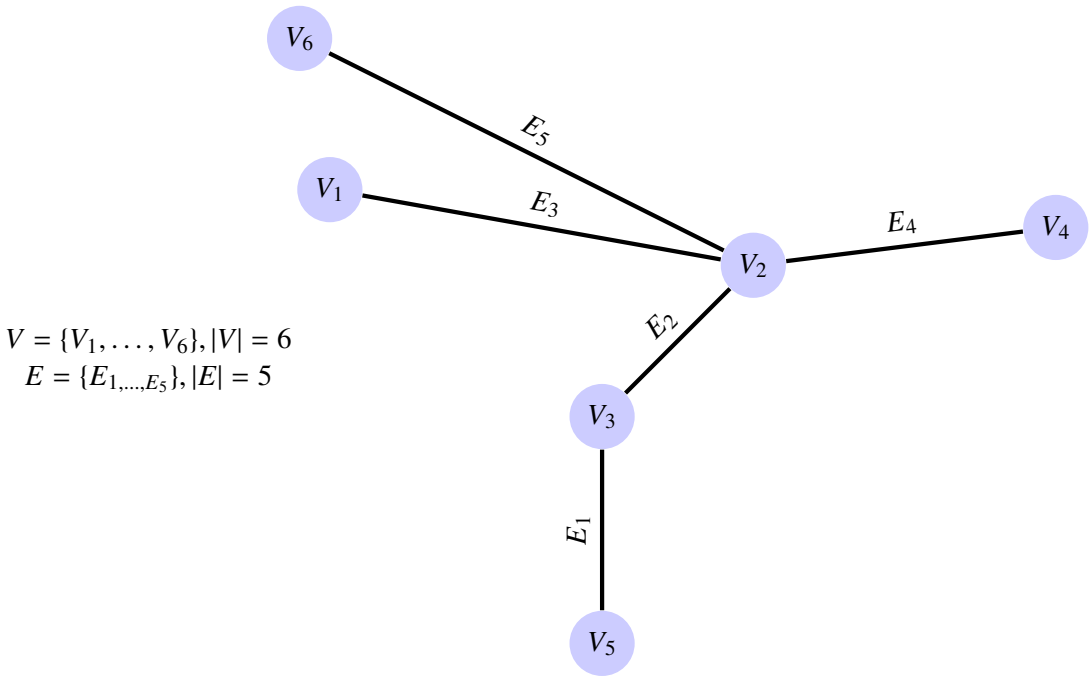
Definice 1.5.8. (Strom) Pojmem **strom** rozumíme souvislý graf neobsahující kružnici.

Teorém 1.5.9. (Alternativní definice) Nechť je dán neorientovaný graf G. Následující trvzení jsou ekvivalentní:

1. G je strom.
2. Každé dva vrcholy v G jsou spojeny právě jednou cestou.
3. G je souvislý, avšak po odebrání jakékoli hrany se G stane nesouvislým.
4. G neobsahuje kružnici, ale po přidaní jakékoli hrany se G stane cyklickým.
5. G je souvislý a platí $|V| = |E| + 1$.

Definice 1.5.10. (Les) Množinu navzájem nepropojených stromů nazveme **les**. Též hovoříme o neorientovaném grafu, ve kterém jsou libovolné dva vrcholy spojeny nejvýše jednou cestou.

Definice 1.5.11. (Polystrom) Pokud nahradíme orientované strany neorientovanými, a tím získáme neorientovaný acyklický graf, jedná se o **polystrom**.



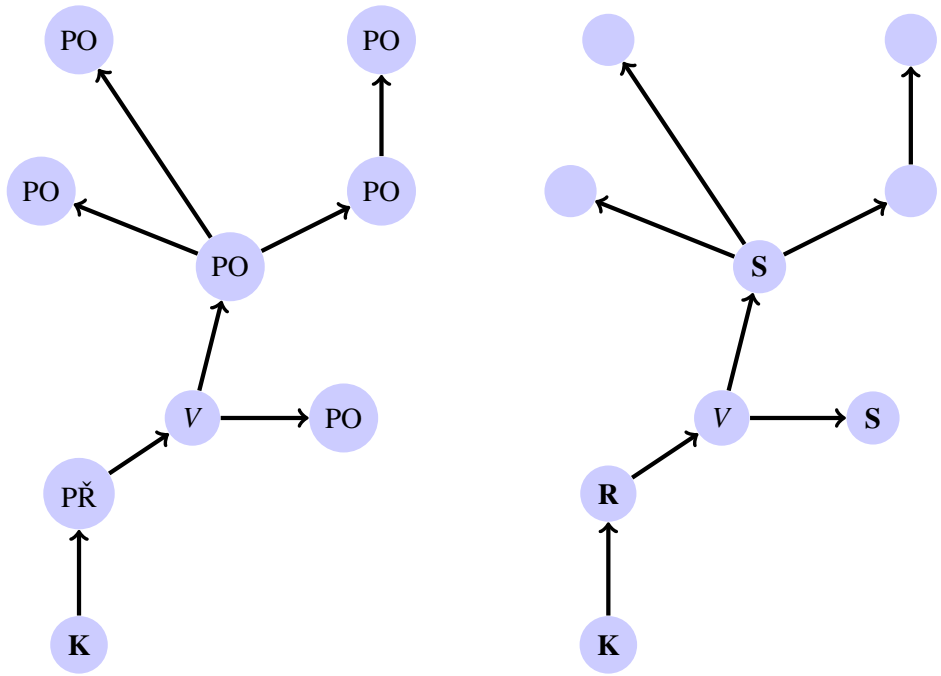
Obrázek 1.7: Ukázka neorientovaného stromu.

Pro nás je ovšem důležité se v grafu orientovat, respektive znát směr. Rozšiřme tedy neorientované stromy o jeden význačný vrchol, a to *kořen*. Tím strom zakořeníme a definujeme v něm orientaci hran. Nejprve však nastiňme základní názvosloví pro lepší orientaci v pozdější látce. Pokud nalezneme vrchol stromu takový, že z něho nevede žádná orientovaná hrana, nazveme tento vrchol *listem*.

Definice 1.5.12. (Názvosloví) Nechť je dán strom G s jedním kořenem K a V je nějaký jeho vrchol mimo kořen a list. Platí:

- Vrcholům se též říká **uzly**.
- Uzel ležící na cestě z uzlu V do libovolného listu stromu se nazývá **potomek** V (PO).
- **Předek** V je libovolný uzel na jednoznačné cestě od kořene do uzlu V (PŘ).
- **List** je uzel, který nemá žádné potomky.
- **Větev** označuje jednoznačně určenou cestu od kořene k listu.
- Bezprostředně nadcházející uzel V ve směru kořene se nazývá **syn** (S) uzlu V.
- Uzel bezprostředně předcházející V je **rodič** (R) uzlu V.

Poznámka. Kořen stromu tedy nemá rodiče a list stromu nemůže mít žádné syny. Ostatní uzly mohou mít libovolný počet synů. Kořen stromu je právě jeden.



Obrázek 1.8: Ukázka orientovaného stromu a vztahu mezi uzly z definice 1.5.12.

1.6 Stromové struktury

Nyní můžeme i díky sekci 1.5 zadefinovat nový typ datové struktury, a to v praxi hojně využívané *stromové struktury* (dále jen **stromy**). Dosud jsme na *stromy* nahlíželi spíše matematicky. V této sekci budeme chtít nastínit jejich potenciál v oblasti reprezentace dat pomocí těchto datových struktur.

Definice 1.6.1. (Strom jako datová struktura) **Stromem** rozumíme abstraktní datový typ, který simuluje hierarchické stromové struktury popsané v podsekci 1.5.1.

Definice 1.6.2. (Podstrom) **Podstromem** nazveme část stromové datové struktury tvořené jedním uzlem a všemi jeho potomky.

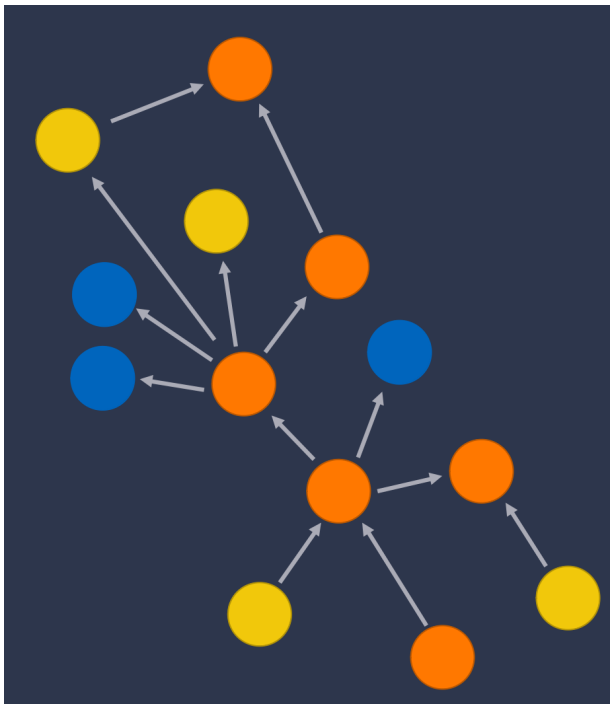
Pomocí stromů dokážeme popsat data, která chceme vysvětlit. Vyvstává otázka, jak tuto strukturu převést na vektor čísel, se kterým jsme již zvyklí pracovat. K tomu bude potřeba jistá parametrická transformační funkce, jejíž parametry se budeme chtít učit (více v kapitole 3).

Stromy dělíme na:

1. neuspořádané – pro daný uzel **nejsou** uspořádáni potomci,
2. uspořádané (zakořeněné) – všichni potomci daného uzlu **jsou** seřazeni, pro naše účely užitečnější, abychom věděli, kde se právě v daném stromu nacházíme.

1.6.1 Uspořádané stromy

My se v rámci této práce omezíme pouze na uspořádané stromy, ve kterých chceme znát návaznost uzlů a vztahy mezi nimi, jelikož v každém uzlu stromu může být obsažena určitá hodnota, podmínka, náhodná veličina nebo další strom. Případně může reprezentovat další oddělenou strukturu dat (např. neuronovou síť). Uspořádané stromy můžeme též označit jako orientované.



Obrázek 1.9: Ukázka uspořádané stromové struktury (polystrom).

Kapitola 2

Základní použití pojmů

V kapitole 1 jsme definovali důležité pojmy z různých oblastí matematiky a optimalizace. Především právě na těchto dvou obsáhlých tématech stojí strojové učení (*ML*[1]), které úzce souvisí s pojmem umělé inteligence (*AI*[2]). Definujme:

Definice 2.0.1. (Artificial Intelligence) Umělá inteligence je oblast informatiky zabývající se tvořením strojů schopných provádět úkony, které typicky vyžadují lidskou inteligenci.

Definice 2.0.2. (Machine Learning) Strojové učení je aplikace umělé inteligence, která umožňuje systémům se učit a vylepšovat se na základě zkušeností bez dodatečného programování. Proces učení probíhá na dodaných datech optimalizací příslušných parametrů k určování predikcí nebo samostatných rozhodnutí.

Strojové učení dělíme na:

- **Učení s učitelem** (*Supervised Learning*) – známe jak výstupu, tak vstupy. Učení probíhá na dodaném datasetu ve snaze predikovat další hodnotu či třídu.
- **Učení bez učitele** (*Unsupervised Learning*) – učící proces probíhá bez dispozice kritérií na správnost hledané transformace. Učení probíhá pouze na informacích ve vstupních dat.
- **Kombinaci výše zmíněných** (*Semi–supervised Learning*) – při procesu učení je k dispozici malé množství vstupů s přiřazenými výstupy a větší množství těch, které je přiřazené nemají.
- **Posilované učení** (*Reinforcement Learning*) – staví na přítomnosti tzv. *agentů* při procesu učení, kteří se učí získáváním odměn či trestů. Není potřeba učitele [21].

V rámci této bakalářské práce se setkáme pouze s prvním ze zmíněných. Zejména při úlohách regrese (již v kapitole 1) a klasifikace (bude vyloženo závěrem této kapitoly). Učení bez učitele se většinou aplikuje při úlohách shlukování, kdy je potřeba zařadit objekt do skupiny s podobnými vlastnostmi.

[1] z anglického *Machine Learning*
[2] z anglického *Artificial Intelligence*

2.1 Analyticky řešitelný příklad Elbo

Ačkoli v praxi používaná pravděpodobnostní rozdělení v kombinaci s obtížně řešitelnými integrály zpravidla nejdou analyticky spočítat a je třeba použít numerický výpočet, ukážeme si v této sekci i jedno z mála analytických řešení jednoduchého, demonstrativního příkladu. Rádi bychom určili, jak vypadá $p(\theta, \alpha|y_1, y_2)$, tedy *aposteriorní* pravděpodobnostní rozdělení pro parametry.

Mějme opět nám již známý systém rovnic $\mathbf{y} = \mathbb{X}\boldsymbol{\theta} + \mathbf{e}$. Nechť dále $\mathbf{y} \in \mathbf{R}^2$, $\boldsymbol{\theta} = (\theta, \theta)^T \in \mathbf{R}^2$, $\mathbf{e} \in \mathbf{R}^2$ a $\mathbb{X} = \mathbb{I} \in \mathbf{R}^{2,2}$. Náš model tedy vypadá takto:

$$\begin{aligned} y_1 &= \theta + e_1 \\ y_2 &= \theta + e_2. \end{aligned} \tag{2.1}$$

Prvně si nadefinujme a označme potřebné členy k výpočtu. Spolu s tabelovanými hodnotami rozdělení zanesme vše do přehledné tabulky:

Zvolená rozdělení	Tabelované hodnoty
$q(\theta) = \mathcal{N}(\mu, \sigma)$	$\mathsf{E}\left(\frac{1}{\alpha}\right) = \frac{\gamma}{\delta}$
$q(\alpha) = \imath\Gamma(\gamma, \delta)$	$\mathsf{E}\left(\ln \alpha\right) = \ln \delta - \psi\left(\gamma\right)$
$y_i \sim \mathcal{N}(\theta, 1), \forall i \in \{1, 2\}$	$\psi(z) = \frac{d}{dz} \ln \Gamma(z)$
$\theta \sim \mathcal{N}(0, \alpha)$	$\imath\Gamma(\alpha, \beta) \propto x^{-\alpha-1} \exp(\beta/x)$
$\alpha \sim \imath\Gamma(0, 0)$	$\imath\Gamma(0, 0) \propto 1/\alpha$

Tabulka 2.1: Přehled parametrů a rozdělení k výpočtu

Poznámka. Ačkoli není Γ–funkce v bodě 0 definována (integrál je divergentní), v praxi definujeme tzv. **Jeffreyho apriorno** [11], kdy pokládáme α a β rovno číslům velice blízkým nule. Jeho výjimečnou vlastností je to, že do systému nevkládáme nadbytečnou informaci, která by mohla zvýšit naši míru neurčitosti. Pro naše účely (a hlavně v praxi) je tedy možné zavést tabelované hodnoty v Tabulce 2.1.

Výpočet

Přepišme hledanou distribuci pomocí znalostí ze sekce 1.1:

$$p(\alpha, \theta|y_1, y_2) \overset{1.1}{=} \frac{p(\alpha, \theta, y_1, y_2)}{p(y_1, y_2)} \overset{1.2}{=} \frac{p(y_1|\theta)p(y_2|\theta)p(\theta)p(\alpha)}{p(y_1, y_2)}. \tag{2.2}$$

Vypočíst čitatele v (2.2) je triviální záležitost dosazení konkrétních distribucí z Tabulky 2.1. Jmenovatele vypočteme marginalizací přes θ a α:

$$\begin{aligned} p(y_1, y_2) = \iint p(\alpha, \theta, y_1, y_2) d\theta d\alpha = \iint p(y_1|\theta)p(y_2|\theta)p(\theta)p(\alpha) d\theta d\alpha \propto \\ \propto \iint \exp\left(-\frac{1}{2}(y_1 - \theta)^2\right) \exp\left(-\frac{1}{2}(y_2 - \theta)^2\right) \frac{1}{\sqrt{\alpha}} \exp\left(-\frac{1}{2\alpha}(0 - \theta)^2\right) \frac{1}{\alpha} d\theta d\alpha \end{aligned} \tag{2.3}$$

Integrál v (2.3) je, bohužel pro nás, analyticky neřešitelný. Tato skutečnost je demonstrována dosazením distribucí bez příslušných normalizačních konstant. Proto výhodně využijeme *KL divergence* mezi aproximativní, navolenou, a skutečnou distribucí, kterou hledáme.

$$KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{y})) \overset{1.47}{=} \int q(\mathbf{z}) \ln\left(\frac{p(\mathbf{z}|\mathbf{y})}{q(\mathbf{z})}\right) d\mathbf{z} \overset{1.9}{=} \iint_H q(\theta)q(\alpha) \ln\left(\frac{p(\theta,\alpha|\mathbf{y})}{q(\theta)q(\alpha)}\right) d\alpha d\theta \overset{ozn.}{=} \mathsf{K} \tag{2.4}$$

Díky vhodně zvoleným aproximativním distribucím lze tento integrál analyticky spočítat. Integrovat budeme přes oba supporty volených distribucí, tedy $\operatorname{supp} q(\theta)$ a $\operatorname{supp} q(\alpha)$. V takovém případě označme H jako kartézský součin těchto dvou množin, tj. $H = \operatorname{supp} q(\theta) \times \operatorname{supp} q(\alpha)$.

$$\mathsf{K} = \iint_H q(\theta)q(\alpha) \ln\left(\frac{p(\theta,\alpha|\mathbf{y})}{q(\theta)q(\alpha)}\right) d\alpha d\theta \overset{1.5}{=} \iint_H q(\theta)q(\alpha) \underbrace{\ln\left(\frac{p(y_1|\theta)p(y_2|\theta)p(\theta)p(\alpha)}{q(\theta)q(\alpha)p(y_1)p(y_2)}\right)}_{\star} d\alpha d\theta$$

Zaměřme se nyní na úpravu logaritmu v integrálu označeného jako $\star$, do které dosaďme patřičné hustoty pravděpodobnosti. Výraz se budeme snažit co nejvíce zjednodušit a zpřehlednit.

$$\star = \Big(\ln\big(p(y_1|\theta)\big) + \ln\big(p(y_2|\theta)\big) + \ln\big(p(\theta)\big) + \ln\big(p(\alpha)\big) - \ln\big(q(\theta)\big) - \ln\big(q(\alpha)\big) - \ln\big(p(y_1)\big) - \ln\big(p(y_2)\big)\Big) \Rightarrow$$

$$\begin{aligned}\mathsf{K} = &\iint_H q(\theta)q(\alpha) \ln\big(p(y_1|\theta)\big) d\alpha d\theta + \iint_H q(\theta)q(\alpha) \ln\big(p(y_2|\theta)\big) d\alpha d\theta + \iint_H q(\theta)q(\alpha) \ln\big(p(\theta)\big) d\alpha d\theta + \\ &+ \iint_H q(\theta)q(\alpha) \ln\big(p(\alpha)\big) d\alpha d\theta - \iint_H q(\theta)q(\alpha) \ln\big(q(\theta)\big) d\alpha d\theta - \iint_H q(\theta)q(\alpha) \ln\big(q(\alpha)\big) d\alpha d\theta - \\ &- \iint_H q(\theta)q(\alpha) \ln\big(p(y_1)\big) d\alpha d\theta - \iint_H q(\theta)q(\alpha) \ln\big(p(y_2)\big) d\alpha d\theta = \diamond\end{aligned}$$

Díky znalosti vlastností z kapitoly 1 teď výhodně na všechny dílčí integrály aplikujeme Fubiniho[3] větu a vypočteme. Prvně je rozdělme na tři skupiny:

- **1. skupina**: Analyticky řešitelné integrály.

$$\begin{aligned}\iint_H q(\theta)q(\alpha) \ln\big(p(y_1|\theta)\big) d\alpha d\theta &= \mathsf{E}_{q(\theta)}\Big[\ln\big(p(y_1|\theta)\big)\Big] \propto \left\langle -\frac{1}{2}(y_1-\theta)^2 \right\rangle \\ \iint_H q(\theta)q(\alpha) \ln\big(p(y_2|\theta)\big) d\alpha d\theta &= \mathsf{E}_{q(\theta)}\Big[\ln\big(p(y_2|\theta)\big)\Big] \propto \left\langle -\frac{1}{2}(y_2-\theta)^2 \right\rangle \\ \iint_H q(\theta)q(\alpha) \ln\big(p(\theta)\big) d\alpha d\theta &= \mathsf{E}_{q(\theta)}\Big[\ln\big(p(\theta)\big)\Big] \propto \left\langle -\frac{\theta^2}{2\alpha} + \ln\left(\frac{1}{\sqrt{\alpha}}\right) \right\rangle \\ \iint_H q(\theta)q(\alpha) \ln\big(p(\alpha)\big) d\alpha d\theta &= \mathsf{E}_{q(\alpha)}\Big[\ln\big(p(\alpha)\big)\Big] \propto \left\langle \ln\left(\frac{1}{\alpha}\right) \right\rangle\end{aligned} \tag{2.5}$$

- **2. skupina**: Tabulkové entropie příslušných rozdělení.

$$\begin{aligned}\iint_H q(\theta)q(\alpha) \ln\big(q(\theta)\big) d\alpha d\theta &= \int_{\operatorname{supp} q(\theta)} \ln\big(q(\theta)\big) q(\theta) \left(\int_{\operatorname{supp} q(\alpha)} q(\alpha) d\alpha\right) d\theta = -\mathsf{H}[q(\theta)] \\ \iint_H q(\theta)q(\alpha) \ln\big(q(\alpha)\big) d\alpha d\theta &= \int_{\operatorname{supp} q(\alpha)} \ln\big(q(\alpha)\big) q(\alpha) \left(\int_{\operatorname{supp} q(\theta)} q(\theta) d\theta\right) d\alpha = -\mathsf{H}[q(\alpha)]\end{aligned} \tag{2.6}$$

[3] *Guido Fubini* (1879–1943)

- **3. skupina**: Hodnoty $p(y_1)$ a $p(y_2)$, jež nejdou analyticky spočítat. Jedná se o člen $\ln p(\mathbf{x})$ v (1.47), o kterém bezpečně víme, že nezávisí na θ, protože integrace přes tento parametr v nich již proběhla.

$$\iint_H q(\theta)q(\alpha)\ln\Big(p(y_1)\Big)d\alpha d\theta = \ln\Big(p(y_1)\Big)\int_{\operatorname{supp} q(\theta)} q(\theta)\left(\int_{\operatorname{supp} q(\alpha)} q(\alpha)d\alpha\right)d\theta = \ln\Big(p(y_1)\Big)$$

$$\iint_H q(\theta)q(\alpha)\ln\Big(p(y_2)\Big)d\alpha d\theta = \ln\Big(p(y_2)\Big)\int_{\operatorname{supp} q(\theta)} q(\theta)\left(\int_{\operatorname{supp} q(\alpha)} q(\alpha)d\alpha\right)d\theta = \ln\Big(p(y_2)\Big)$$

Poznámka. **3. skupinu** označíme celkově za *konstantu* c, která neovlivní numerickou optimalizaci a je možné ji vzhledem k K zanedbat.

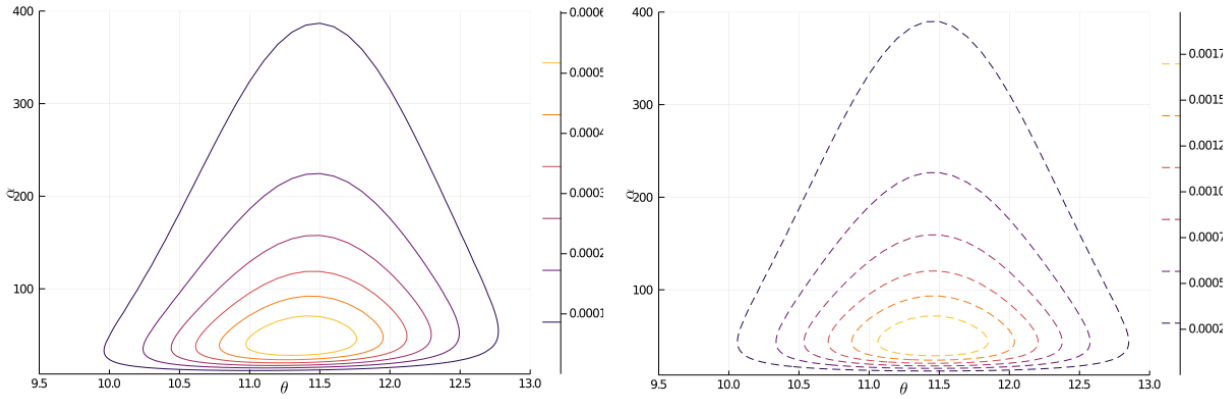
$$\diamond = \mathsf{E}_{q(\theta)}\Big[\ln\Big(p(y_1|\theta)\Big)\Big] + \mathsf{E}_{q(\theta)}\Big[\ln\Big(p(y_2|\theta)\Big)\Big] + \mathsf{E}_{q(\theta)}\Big[\ln\Big(p(\theta)\Big)\Big] + \mathsf{E}_{q(\alpha)}\Big[\ln\Big(p(\alpha)\Big)\Big] + \mathsf{H}\left[q(\theta)\right] + \mathsf{H}[q(\alpha)] + c \tag{2.7}$$

K jsme nakonec upravili do podoby (2.7). Jelikož jsou v optimalizačních procesech optimalizovány parametry, není přímo nutné pracovat s normalizačními konstantami, které na nich nezávisí. Proto dosadíme (2.5) do (2.7) a spočítáme:

$$\begin{aligned}\mathsf{K} &\propto \left\langle -\frac{1}{2}(y_1-\theta)^2\right\rangle + \left\langle -\frac{1}{2}(y_2-\theta)^2\right\rangle + \left\langle -\frac{\theta^2}{2\alpha} + \ln\left(\frac{1}{\sqrt{\alpha}}\right)\right\rangle + \left\langle \ln\left(\frac{1}{\alpha}\right)\right\rangle + \mathsf{H}\left[q(\theta)\right] + \mathsf{H}[q(\alpha)] \propto \\ &\propto -\frac{1}{2}\left(y_1^2 - 2y_1\mu + \mu^2 + \sigma\right) - \frac{1}{2}\left(y_2^2 - 2y_2\mu + \mu^2 + \sigma\right) - \frac{1}{2}\left(\ln(\delta) - \psi(\gamma) + \frac{\gamma}{\delta}(\mu^2+\sigma)\right) + \\ &+ \psi(\gamma) - \ln(\delta) + \left(\frac{1}{2}\ln(\sigma)\right) + \Big(\gamma + \ln(\delta\Gamma(\gamma)) - (1+\gamma)\psi(\gamma)\Big).\end{aligned} \tag{2.8}$$

Tím jsme spočetli hodnotu *KL divergence* závisející na čtveřici parametrů $(\mu, \sigma, \gamma, \delta)$, která může být nyní naimplementována do programovacího prostředí a optimalizována přes zmíněné parametry, tzn. hledáme

$$(\hat{\mu}, \hat{\sigma}, \hat{\gamma}, \hat{\delta}) = \underset{\mu,\sigma,\gamma,\delta}{\arg\min}\, KL\left(q(\theta,\alpha|\mu,\sigma,\gamma,\delta)||p(\theta,\alpha|y_1,y_2)\right). \tag{2.9}$$



Obrázek 2.1: Contour plot distribuce $p(\theta, \alpha|y_1, y_2)$ (vlevo) vyčíslené v bodech $(y_1, y_2) = (11, 12)$ a distribuce $q(\theta, \alpha|\mu, \sigma, \gamma, \delta)$ v hodnotách odhadu $\hat{\mu}, \hat{\sigma}, \hat{\gamma}, \hat{\delta}$.

2.2 Analyticky řešitelný vícerozměrný příklad Elbo

Uvažujme vícedimenzionální pravděpodobnostní model s následujícími pravděpodobnostními distribucemi:

$$\begin{aligned} p(\mathbf{y}, \boldsymbol{\theta}|\mathbb{X}, \alpha) &= p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X})p(\boldsymbol{\theta}|\alpha) = \mathcal{N}(\mathbb{X}\boldsymbol{\theta}, \mathbb{I})\mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbb{I}) \\ p(\boldsymbol{\theta}|\alpha, \mathbf{y}, \mathbb{X}) &\propto p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X})p(\boldsymbol{\theta}|\alpha) = \exp\left(-\frac{1}{2}\left\|\mathbf{y} - \mathbb{X}\boldsymbol{\theta}\right\|_2^2 - \frac{1}{2}\alpha\left\|\boldsymbol{\theta}\right\|_2^2\right). \end{aligned} \tag{2.10}$$

Poznámka. Je dobré zamyslet se zde nad významem symbolu α. Matematicky se jedná o *skalár*, ovšem můžeme si ho také představit jako konstantu násobící $(d+1)$–dimenzionální vektor jedniček, tj. $\alpha \cdot (1, \ldots, 1)^T = (\alpha, \ldots, \alpha)^T$. Pro všechny případy je tedy α stejná. Díky užité konvenci zapisujeme jako α^{-1} před kovarianční matici kvůli přehlednějšímu zápisu vícedimenzionálního Gaussova rozdělení.

I přes předpoklad znalosti řešení z (1.46),

$$p(\boldsymbol{\theta}|\alpha, \mathbf{y}, \mathbb{X}) \propto \mathcal{N}\left(\boldsymbol{\theta}|(\mathbb{X}^T\mathbb{X} + \alpha\mathbb{I})^{-1}\mathbb{X}^T\mathbf{y}, (\mathbb{X}^T\mathbb{X} + \alpha\mathbb{I})^{-1}\right),$$

ho zkusíme odvodit použitím minimalizace KL divergence. Abychom mohli zavedenou KL diveregenci použít, je třeba předpokládat aproximativní distribuci $q(\boldsymbol{\theta}) = \mathcal{N}(\hat{\boldsymbol{\theta}}, \Sigma)$.

$$\begin{aligned} KL(q||p) &= \int q(\boldsymbol{\theta}) \ln\left\{\frac{q(\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X})p(\boldsymbol{\theta})}\right\} d\boldsymbol{\theta} = \int q(\boldsymbol{\theta})\left(\ln q(\boldsymbol{\theta}) - \ln p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X}) - \ln p(\boldsymbol{\theta})\right) d\boldsymbol{\theta} = \\ &= \int q(\boldsymbol{\theta}) \ln q(\boldsymbol{\theta}) d\boldsymbol{\theta} + \int q(\boldsymbol{\theta})\left(-\ln p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X}) - \ln p(\boldsymbol{\theta})\right) d\boldsymbol{\theta} \end{aligned} \tag{2.11}$$

V rovnici (2.11) rozeznáváme již známou definovanou veličinu (1.21).

2.2.1 Entropie vícerozměrného Gaussova rozdělení

Dále si připomeňme, jak vypadá vícerozměrné Gaussovo pravděpodobnostní rozdělení náhodného vektoru $\boldsymbol{\theta}$ (1.26):

$$p(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d+1}{2}}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right\} \propto \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right\},$$

kde $d \in \mathbf{N}$, $\hat{\boldsymbol{\theta}}$ je $(d+1)$–dimenzionální vektor středních hodnot, $\boldsymbol{\Sigma}$ kovarianční matice dimenze $(d+1) \times (d+1)$ a $|\boldsymbol{\Sigma}|$ označuje hodnotu determinantu matice $\boldsymbol{\Sigma}$. Spočtěme tedy hodnotu entropie:

$$\begin{aligned} \int q(\boldsymbol{\theta}) \ln q(\boldsymbol{\theta}) d\boldsymbol{\theta} &= -\left(-\int q(\boldsymbol{\theta}) \ln q(\boldsymbol{\theta}) d\boldsymbol{\theta}\right) \propto \\ &\propto -\left(-\int q(\boldsymbol{\theta}) \ln\left(\frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\Sigma^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right\}\right) d\boldsymbol{\theta}\right) \propto \\ &\propto -\left(-\int q(\boldsymbol{\theta}) \ln\left(\frac{1}{|\Sigma|^{1/2}}\right) d\boldsymbol{\theta} + \frac{1}{2}\int q(\boldsymbol{\theta})\left((\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\Sigma^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right) d\boldsymbol{\theta}\right) \propto \\ &\propto -\left(-\mathsf{E}_{q(\boldsymbol{\theta})}\left[\ln\left(\frac{1}{|\Sigma|^{1/2}}\right)\right] + \frac{1}{2}\mathsf{E}_{q(\boldsymbol{\theta})}\left[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\Sigma^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right]\right) \propto \\ &\propto -\left(-\ln\left(\frac{1}{|\Sigma|^{1/2}}\right) + 0\right) \propto \\ &\propto -\frac{1}{2}\ln\left(|\Sigma|\right). \end{aligned} \tag{2.12}$$

Výpočet KL divergence a řešení

Nyní je třeba dopočítat vztah (2.11). První člen jsme již spočetli a vyjádřili jako zápornou *spojitou entropii* vícedimenzionálního Gaussova pravděpodobnostního rozdělení. V druhém členu pozorujeme definovaný výraz střední hodnoty argumentu při míře $q(\boldsymbol{\theta})$. Zřejmě platí i Fubiniho věta díky omezenosti a měřitelnosti pravděpodobnostních funkcí.

$$\int q(\boldsymbol{\theta}) \left(-\ln p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X}) - \ln p(\boldsymbol{\theta})\right) d\boldsymbol{\theta} = \mathsf{E}_{q(\boldsymbol{\theta})} \left[-\ln p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X}) - \ln p(\boldsymbol{\theta})\right] \overset{ozn.}{=}$$

$$\overset{ozn.}{=} \left\langle \frac{1}{2} \left((\mathbf{y} - \mathbb{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbb{X}\boldsymbol{\theta}) + \alpha \boldsymbol{\theta}^T \boldsymbol{\theta} \right) \right\rangle = \left\langle \frac{1}{2} \left(\mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T \mathbb{X}^T \mathbf{y} - \mathbf{y}^T \mathbb{X} \boldsymbol{\theta} + \boldsymbol{\theta}^T \mathbb{X}^T \mathbb{X} \boldsymbol{\theta} + \alpha \boldsymbol{\theta}^T \boldsymbol{\theta} \right) \right\rangle =$$

Než budeme počítat dál, je velice výhodné uvědomit si, co vlastně člen $\boldsymbol{\theta}^T \mathbb{X}^T \mathbb{X} \boldsymbol{\theta}$ rozměrově reprezentuje. Ověřme rozměry: $\boldsymbol{\theta}^T \in \mathbf{R}^{1 \times d+1}$, $\mathbb{X}^T \in \mathbf{R}^{d+1 \times n}$, $\mathbb{X} \in \mathbf{R}^{n \times d+1}$ a $\boldsymbol{\theta} \in \mathbf{R}^{d+1 \times 1}$. Po vynásobení všemi komponenty vznikne skalár, tedy regulární matice 1×1, na kterou můžeme beztrestně aplikovat lineární maticový operátor Tr – *Stopa*[4]. Stopa nám nabízí možnost prohození pořadí $\boldsymbol{\theta}^T$ v jejím argumentu (při neporušení rozměrové podmínky násobení). Souběžně aplikujme tuto úpravu i na poslední člen.

$$\begin{aligned}
&= \left\langle \frac{1}{2} \left(\mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T \mathbb{X}^T \mathbf{y} - \mathbf{y}^T \mathbb{X} \boldsymbol{\theta} + \mathrm{Tr}\left(\boldsymbol{\theta}^T \mathbb{X}^T \mathbb{X} \boldsymbol{\theta} \right) + \mathrm{Tr}\left(\alpha \boldsymbol{\theta}^T \boldsymbol{\theta} \right) \right) \right\rangle = \\
&= \left\langle \frac{1}{2} \left(\mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T \mathbb{X}^T \mathbf{y} - \mathbf{y}^T \mathbb{X} \boldsymbol{\theta} + \mathrm{Tr}\left(\mathbb{X}^T \mathbb{X} \boldsymbol{\theta} \boldsymbol{\theta}^T \right) + \mathrm{Tr}\left(\alpha \boldsymbol{\theta} \boldsymbol{\theta}^T \right) \right) \right\rangle = \\
&= \left\langle \frac{1}{2} \left(\mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T \mathbb{X}^T \mathbf{y} - \mathbf{y}^T \mathbb{X} \boldsymbol{\theta} + \mathrm{Tr}\left(\mathbb{X}^T \mathbb{X} \boldsymbol{\theta} \boldsymbol{\theta}^T + \alpha \boldsymbol{\theta} \boldsymbol{\theta}^T \right) \right) \right\rangle = \\
&= \left\langle \frac{1}{2} \left(\mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T \mathbb{X}^T \mathbf{y} - \mathbf{y}^T \mathbb{X} \boldsymbol{\theta} + \mathrm{Tr}\left((\mathbb{X}^T \mathbb{X} + \alpha \mathbb{I}) \boldsymbol{\theta} \boldsymbol{\theta}^T \right) \right) \right\rangle
\end{aligned} \tag{2.13}$$

V posledním kroce aplikujeme střední hodnotu a její vlastnosti na výraz (2.13) při míře $q(\boldsymbol{\theta})$. Tedy:

$$\frac{1}{2} \left(\mathbf{y}^T \mathbf{y} - \hat{\boldsymbol{\theta}}^T \mathbb{X}^T \mathbf{y} - \mathbf{y}^T \mathbb{X} \hat{\boldsymbol{\theta}} + \mathrm{Tr}\left((\mathbb{X}^T \mathbb{X} + \alpha \mathbb{I}) (\hat{\boldsymbol{\theta}} \hat{\boldsymbol{\theta}}^T + \boldsymbol{\Sigma}) \right) \right). \tag{2.14}$$

Nyní jsme schopni vyjádřit výslednou $KL(q||p)$ jako součet (2.12) a (2.14),

$$KL(q||p) = -\frac{1}{2} \ln\left(|\boldsymbol{\Sigma}|\right) + \frac{1}{2} \left(\mathbf{y}^T \mathbf{y} - \hat{\boldsymbol{\theta}}^T \mathbb{X}^T \mathbf{y} - \mathbf{y}^T \mathbb{X} \hat{\boldsymbol{\theta}} + \mathrm{Tr}\left((\mathbb{X}^T \mathbb{X} + \alpha \mathbb{I}) (\hat{\boldsymbol{\theta}} \hat{\boldsymbol{\theta}}^T + \boldsymbol{\Sigma}) \right) \right), \tag{2.15}$$

a optimalizovat numericky.

[4] z anglického *Trace*

2.3 Metoda logistické regrese

Logistickou regresí označujeme metodu, díky níž dokážeme jisté třídě nebo jevu existující ve dvou stavech přiřadit pravděpodobnost v intervalu $[0, 1]$. Lze rozšířit na více tříd nebo jevů. Díky znalostí ze sekce 1.4 lze logistickou regresi označit za jednovrstvou neuronovou síť se sigmoidální aktivační funkcí.

Nechť $\mathbf{y}$ je vektor n čísel složených pouze z nul a jedniček (například v (2.16)). Rádi bychom nyní tuto binární třídu predikovali. Převeďme tuto myšlenku do řeči pravděpodobnosti:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \xrightarrow{\mathrm{P}} \begin{pmatrix} \mathrm{P}(\mathrm{Y}_1 = 1|\Phi(\mathbf{x}_1)) \\ \mathrm{P}(\mathrm{Y}_2 = 0|\Phi(\mathbf{x}_2)) \\ \mathrm{P}(\mathrm{Y}_3 = 1|\Phi(\mathbf{x}_3)) \\ \vdots \\ \mathrm{P}(\mathrm{Y}_n = 0|\Phi(\mathbf{x}_n)) \end{pmatrix} \overset{1.1.3}{=} \begin{pmatrix} p(y_1 = 1|\Phi(\mathbf{x}_1)) \\ p(y_2 = 0|\Phi(\mathbf{x}_2)) \\ p(y_3 = 1|\Phi(\mathbf{x}_3)) \\ \vdots \\ p(y_n = 0|\Phi(\mathbf{x}_n)) \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_n \end{pmatrix}. \tag{2.16}$$

Událost, zda jev nastal nebo ne, se modeluje pomocí náhodné veličiny $\mathbf{Y} = (\mathrm{Y}_1, \ldots, \mathrm{Y}_n)^T$, která ve svých složkách nabývá hodnoty buď 0, kdy jev nenastal, nebo 1, pokud jev nastal. Takové složky náhodné veličiny Y_i se řídí **Bernoulliho** (alternativním) **rozdělením** s parametrem p_i, tj.

$$\mathrm{Y}_i \sim \mathrm{Be}(p_i), \ \forall i \in \hat{n}. \tag{2.17}$$

Rádi bychom však dokázali určit pravděpodobnost, s jakou i–tá složka $\mathbf{Y}$ nabyde hodnotu 0, či 1. K dispozici máme množinu pozorování, odpovídající bázové funkce a cílové hodnoty.

$$\mathrm{Y}_i = \begin{cases} 1, & \text{s pravděpodobností} \quad p_i = p(\mathrm{C}_1|\Phi(\mathbf{x}_i)) \\ 0, & \text{s pravděpodobností} \quad 1 - p_i = 1 - p(\mathrm{C}_1|\Phi(\mathbf{x}_i)) = p(\mathrm{C}_2|\Phi(\mathbf{x}_i)) \end{cases} \tag{2.18}$$

Poznámka. C_1 a C_2 zde označuje pojem **třídy** (*class*), konkrétně třídy jedniček a nul (někdy také *pozitivní* a *negativní* případy).

2.3.1 Logistická funkce

Pomocí **logistické funkce** (1.56) lze interpretovat transformaci pro i–tou složku náhodné veličiny $\mathbf{Y}$:

$$p_i = p(\mathrm{C}_1|\Phi(\mathbf{x}_i)) = f_\sigma\left(\boldsymbol{\theta}^T \Phi(\mathbf{x}_i)\right). \tag{2.19}$$

Díky jejím vlastnostem,

$$\lim_{x \to -\infty} f_\sigma(x) = \lim_{x \to -\infty} \frac{1}{1 + \exp(-x)} = 0 \ \wedge \lim_{x \to +\infty} f_\sigma(x) = \lim_{x \to +\infty} \frac{1}{1 + \exp(-x)} = 1, \tag{2.20}$$

dostaneme na levé straně (2.19) vždy číslo mezi 0 a 1 (neuvažujeme limitní stavy). Říkáme tedy, že n–rozměrnému vektoru $\mathbf{y}$ odpovídá příslušný vektor pravděpodobností, který ho vysvětluje.

Pokud

$$f_\sigma(x) \begin{cases} \to 1, & \text{pak jsme si jistější jedničkou.} \\ \to 0, & \text{pak jsme si jistější nulou.} \\ = \frac{1}{2}, & \text{pak se nacházíme v oblasti největšího šumu, nedokážeme rozhodnout.} \end{cases}$$

Poznámka. Vektor $\mathbf{y}$ zde reprezentuje realizaci náhodné veličiny s Bernoulliho rozdělením a se sigmoidální pravděpodobnostní funkcí.

Mějme set $S = \{y_i, \Phi(\mathbf{x}_i) \mid i \in \hat{n}\}$, kde $y_i \in \{0, 1\}$, $\Phi(\mathbf{x}_i)$ značí vektor bázových funkcí pro i–té pozorování a $n \in \mathbf{N}$. Zapišme odpovídající hustotu pravděpodobnosti:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{i=1}^{n} (p_i)^{y_i} (1 - p_i)^{1-y_i} \tag{2.21}$$

Ztrátovou funkci $L(\boldsymbol{\theta})$ zapíšeme jako záporný logaritmus (2.21), tj.

$$L(\boldsymbol{\theta}) = -\ln(p(\mathbf{y}|\boldsymbol{\theta})) = -\sum_{i=1}^{n} \Big(y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \Big). \tag{2.22}$$

2.3.2 Bayesovská logistická regrese

Kdybychom chtěli na logistickou regresi nahlížet **bayesovsky**, tedy vyjádřit

$$p(\boldsymbol{\theta}|\mathbf{y}), \tag{2.23}$$

setkáme se s nemožností integrace v rámci Bayesova pravidla. Proto je třeba zavést **reparametrizační trik**, díky nemuž (2.23) aproximujeme a následně použitím Elbo zoptimalizujeme.

Reparametrizační trik

Nechť $e \sim \mathcal{N}(0, 1)$. Pro $q(m) = \mathcal{N}\left(\mu_m, \sigma_m^2\right)$ a nějakou $f(m)$ lze aproximovat:

$$m^{(i)} = \mu_m + \sigma_m^2 e^{(i)}, \quad \forall i \in \hat{n}. \tag{2.24}$$

$$\mathsf{E}_{q(m)}[f(m)] \overset{\text{Monte Carlo}}{\approx} \frac{1}{n} \sum_{i=1}^{n} f\left(m^{(i)}\right) \overset{2.24}{=} \frac{1}{n} \sum_{i=1}^{n} f\left(\mu_m + \sigma_m^2 e^{(i)}\right) \tag{2.25}$$

Toto je přesné pro velká n. Ovšem pokud budeme brát $n = 1$, získáme nestranný (nevychýlený) odhad gradientu. Budeme-li výpočet mnohokrát opakovat, docílíme rovnosti:

$$\mathsf{E}_{q(m)}[f(m)] = \mathsf{E}_{p(e)}\left[f\left(\mu_m + \sigma_m^2 e\right)\right], \tag{2.26}$$

což nám umožní optimalizovat přes parametry μ_m a σ_m^2 a vyhnout se tak problému s integrací v rámci Bayesova pravidla.

2.3.3 Vícetřídová logistická regrese

Pokud bychom chtěli pracovat s k třídami, musíme zavést novou transformační vektorovou funkci. Jedná se o tzv. zobecnění logistické funkce, jejíž vektorový výstup má hodnoty složek omezené na intervalu $(0, 1)$ a jejich součet je 1. Díky této transformaci dokážeme převést vícetřídový problém na problém, který již umíme řešit.

Definice 2.3.1. (Softmax funkce) Vektorové zobrazení $\mathbf{f}_{\boldsymbol{\sigma}} : \mathbf{R}^n \to \mathbf{R}^n$, jehož i–tá složka je definována jako

$$f_{\sigma_i}(\mathbf{x}) = \frac{\exp(x_i)}{\sum_{k=1}^{n} \exp(x_k)}, \ \forall i \in \hat{n}, \tag{2.27}$$

nazveme **softmax** funkce.

Příklad

Generujeme 1000 dvoudimenzionálních pozorování z $\mathcal{N}_2(\mathbf{0}, \mathbb{I})$, jež uspořádáme do matice $\mathbb{X}$. Nechť pravé $\boldsymbol{\theta}'$ je např. $(0, 10)$. Sestavme nyní způsob, jakým chceme generovat vektor $\mathbf{y}'$ pomocí výše zmíněného:

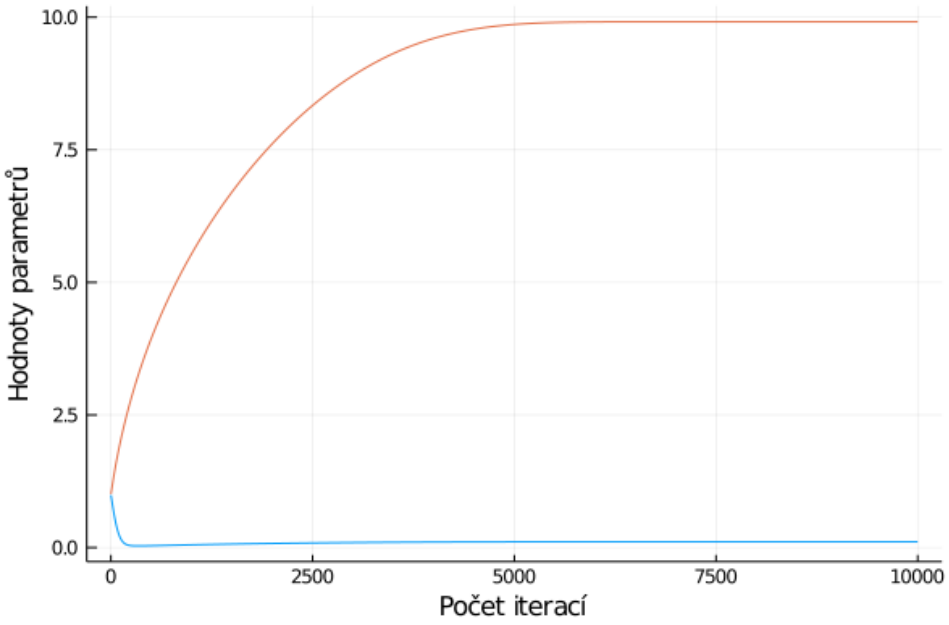
$$\mathbf{y}' = \sigma(\boldsymbol{\theta}'^T \cdot \mathbb{X}) \tag{2.28}$$

Aplikací Bernoulliho rozdělení na vektor $\mathbf{y}'$ získáme vektor $\mathbf{y}$ obsahující pouze 0 nebo 1.

Model můžeme sestavit jako jednovrstvou neuronovou síť se sigmoidální aktivační funkcí, tj.

$$\text{model}(\boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^T \cdot \mathbb{X}), \tag{2.29}$$

s odpovídající ztrátovou funkcí (2.22) a natrénovat. Jako optimalizátor je použit **ADAM** s krokem 0.01. Proces učení $\boldsymbol{\theta}$ začneme v bodě $(1, 1)$ a jako počet iterací zvolíme dostatečných 10000.



Obrázek 2.2: Proces učení parametru $\boldsymbol{\theta}$ v závislosti na počtu iterací.

Na Obrázku 2.2 je vidět, že naučený parametr $\boldsymbol{\theta}$ odpovídá námi zadanému parametru $\boldsymbol{\theta}'$.

Kapitola 3

Vysvětlitelnost

Často se v rámci umělé inteligence a strojového učení setkáme s metodami, které i přes jejich výkonnost a schopnost řešit složité úkoly nedokáží interpretovat dosažené výsledky v jednoduché a srozumitelné podobě. Jeden z důvodů tkví ve zpracovávání vysokodimenzionálních vstupních dat v kombinaci s množstvím nelineárních a do sebe vnořených transformací, díky kterým jsou dosahovány pravděpodobnostní rozhodnutí napříč modelem. Model, jenž takové metody používá, můžeme označit za tzv. *black box* [3]. Název odráží skutečnost, že ani člověk, který model navrhl, nedokáže vysvětlit, proč se AI rozhodla tak, jak se v určitém místě rozhodla. Vysvětlitelnost jde ruku v ruce s pojmem interpretovatelnosti jeho učení, respektive jak určitá rozhodnutí vysvětlit a navíc i interpretovat.

Existuje mnoho možností, jak vysvětlitelnou AI definovat. Jedna z nich je nastíněna v [3]:

Definice 3.0.1. (Explainable AI – XAI) Vysvětlitelná umělá inteligence je systém, který vytváří detaily nebo důvody, proč je jeho fungování jasné nebo snadno srozumitelné.

XAI potřebuje ke svému fungování již zmíněný *black box*. Vytvoří nad ním vrstvu, natrénuje ho a tuto vrstvu zpětně vysvětlí.

Naproti tomu definice interpretovatelného strojového učení [10]:

Definice 3.0.2. (Interpretable ML – IML) Interpretovatelné strojové učení odkazuje na metody a modely, které činí chování a predikce systému pochopitelným pro lidi.

IML se *black boxu* vyhýbá. Od začátku máme model, který má určitý význam a již z principu musí být vysvětlitelný.

Byly též vytyčeny pilíře, jež by měly být cílem XAI. Mezi ně například patří: důvěryhodnost, příčinnost, spolehlivost nebo informativnost modelu. My se v rámci této práci budeme věnovat technikám minimalizace počtu parametrů v modelu k získání tzv. *řídké parametrizace*, která nám umožní model jednodušeji vysvětlit.

3.1 Řídké parametrizace

Představme si, že máme model, který obsahuje tisíce parametrů. Intuitivně můžeme předpokládat, že i kdybychom získali potřebnou přesnost, natrénovat tolik parametrů může zabrat až neúměrně moc času. Proto existují jisté techniky a metody, jakými lze z těchto parametrů vybrat ty, které jsou pro model významnější než ostatní, a díky nimž jej dokážeme patřičně vysvětlit. Tedy, aby co nejvíce z nich bylo nulových za předpokladu, že jsou pro model nevýznamné.

Prvně definujme pojem, se kterým se budeme v této kapitole často setkávat:

Definice 3.1.1. (Příznak) **Příznakem** rozumíme měřitelnou vlastnost nebo charakteristiku pozorovaného jevu. V našich modelech lze ztotožnit s nezávisle proměnnou.

Metody

1. L_0 **regularizace**

   Mějme model s odpovídající ztrátovou funkcí $L(\boldsymbol{\theta})$. Přidejme k ní dodatečnou podmínku na $\boldsymbol{\theta}$, tzv. L_0 normu:

   $$L'(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_0 \text{ , kde } \quad \|\boldsymbol{\theta}\|_0 = \sum_{j=0}^{d} \mathcal{I}\left[w_j \neq 0\right]. \tag{3.1}$$

   Koeficient $\lambda \in \mathbf{R}$ se nazývá váhový regularizační faktor a je možné díky němu regulovat počet nenulových parametrů v modelu. L_0 norma tedy omezuje počet nenulových komponent $\boldsymbol{\theta}$ a napomáhá řídkosti v odhadu $\hat{\boldsymbol{\theta}}$. Řešení optimalizační úlohy (3.1),

   $$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L'(\boldsymbol{\theta}), \tag{3.2}$$

   je však zbytečně složité a výpočetně náročné. Hlavně díky nediferencovatelnosti L_0 normy [8]. Proto budeme využívat jiných technik.

2. **Feature selection**

   Pomocí této metody dokážeme vybrat podmnožinu relevantních příznaků (*features*), které jsou pro model významné. Používáme tři různé přístupy [15]:

   - **Top Down** – postupně snižujeme počet příznaků a pozorujeme, jak se nám mění trénovací čas a přesnost,
   - **Bottom Up** – obdobně jako u Top Down, ovšem s postupným zvyšováním příznaků,
   - **Kombinace Top Down & Bottom Up** – kombinace výše zmíněných, kdy můžeme zpočátku postupně zvyšovat, a poté snižovat počet příznaků. Nebo naopak.

   Pokud již známe řešení, pak tato metoda dokáže ušetřit trénovací čas a získat řídký parametr. Problém tkví v tom, že nám toto řešení nikdo dopředu neřekne. Bez jeho apriorní znalosti je tato metoda výpočetně náročná, a proto ji též používat nebudeme.

3. **Shrinkage prior** – též zvaná *utahující se apriorno*

   Opět uvažujeme $\boldsymbol{\theta}$ jako vektor parametrů. Je pochopitelné, že jen některé jeho komponenty budou daleko od nuly. Například v regresi nebo v klasifikaci s mnoha nezávisle proměnnými očekáváme, že jen některé jsou významné, a proto se pojí s nenulovým parametrem. Z kapitoly 1 víme, že platí:

   $$\text{Aposteriorní distribuce} \propto \text{Věrohodnost} \times \text{Apriorní distribuce} \tag{3.3}$$

   Proto se na problém začneme dívat pravděpodobnostním přístupem a apriorně budeme volit určité distribuce, které nám pomohou zjistit, které parametry jsou v modelu nevýznamné a docílit tak řídké parametrizace modelu [14]. Pokud je nějaký z parametrů v modelu nevýznamný, apriorno převáží a *utáhne* jeho pravděpodobnost k nule.

- **Spike & Slab**

  Tzv. *Spike & Slab prior* [4] je jedno z nejpopulárnějších utahujících se aprioren a často považováno za zlatý standard k docílení řídkého Bayesovského řešení. Jako apriorní distribuce je zvolen mix dvou Gaussových distribucí:

  $$w_j \sim \lambda_j \mathcal{N}(0, c^2) + (1 - \lambda_j)\mathcal{N}(0, \epsilon^2),\ j \in \hat{d}, \tag{3.4}$$

  kde $\epsilon \ll c$ a $\lambda_j \in \{0, 1\}$ řídící se Bernoulliho rozdělením nám říká, zda je parametr w_j blízko nule (*Spike*, implikuje $\lambda_j = 0$) nebo ne (*Slab*, implikuje $\lambda_j = 1$) [14]. U této metody je však nutno nastavit celkem tři parametry $(\lambda, c^2, \epsilon^2)$. Existuje proto jistě nějaké nastavení těchto tří parametrů, kde tato metoda selže.

- **Laplaceova**[1] **distribuce**

  Zvolíme Laplaceovu distribuci [2] s nulovou střední hodnotou jako apriorní, tedy:

  $$p(x|0, b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right). \tag{3.5}$$

  Koeficientem $\frac{1}{b}$ dokážeme korigovat míru, kterou je výraz tlačen k nule (lze pro konzistenci těchto koeficientů označit jako λ). K věrohodnostnímu členu z (3.3) přidáváme exponenciálu s argumentem, jenž se shoduje se záporně vzatou L_1 normou na Euklidovském prostoru [18]:

  $$\exp\left(-\lambda\,|\boldsymbol{\theta}|\right) = \exp\left(-\lambda\,\|\boldsymbol{\theta}\|_1\right) \tag{3.6}$$

  Velká výhoda této apriorní distribuce je konvexnost, která nám v kombinaci s konvexní metodou (např. MNČ) zaručí jedinečnost řešení úlohy.

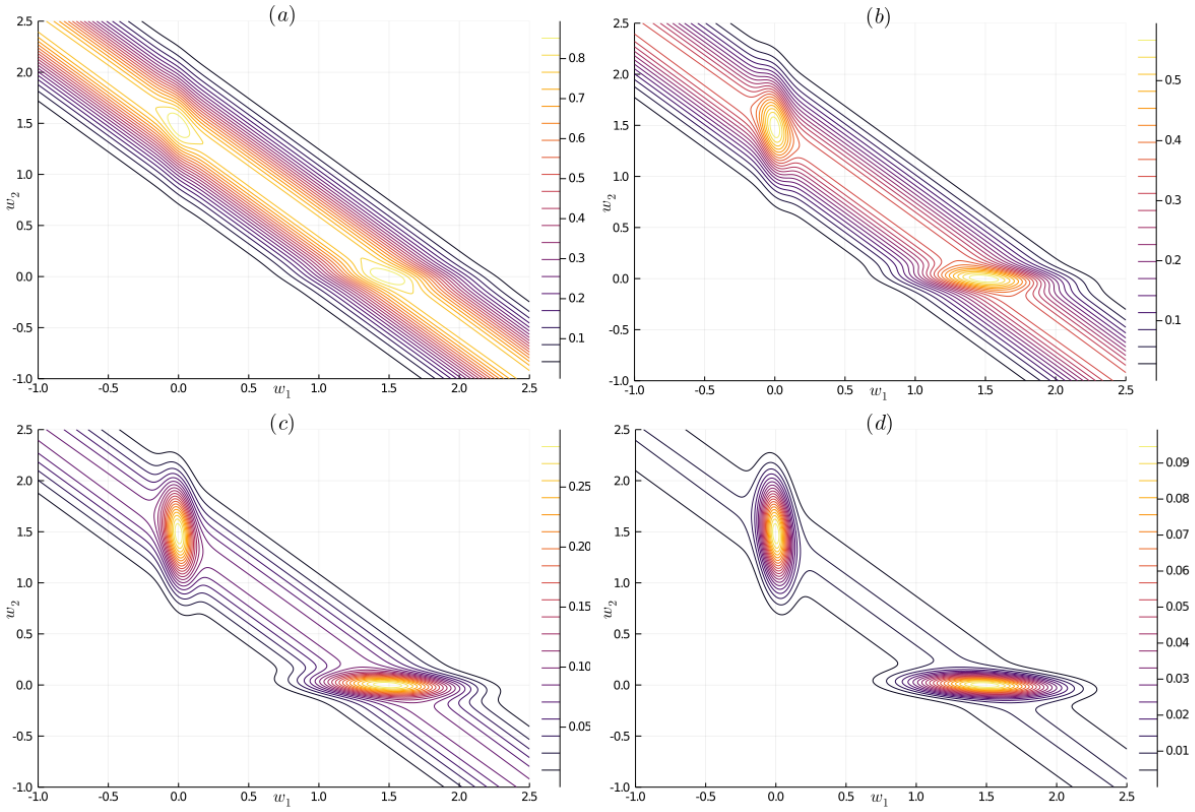
- **Automatic Relevance Determination** (ARD) – jedná se o efektivní nástroj, kterým dokážeme redukovat počet nadbytečných příznaků k řídkému řešení regularizací prostoru řešení za pomocí parametrické apriorní distribuce [22]. Uvažujme jako apriorní distribuci na parametry zobecněné Studentovo[2] rozdělení [16] s nulovou střední hodnotou a dalšími dvěma parametry σ^2 a ν:

  $$p(\boldsymbol{\theta}) = \int p\left(\boldsymbol{\theta}|\boldsymbol{\alpha}\right) p(\boldsymbol{\alpha}) d\boldsymbol{\alpha} = \mathrm{St}(0, \sigma^2, \nu), \tag{3.7}$$

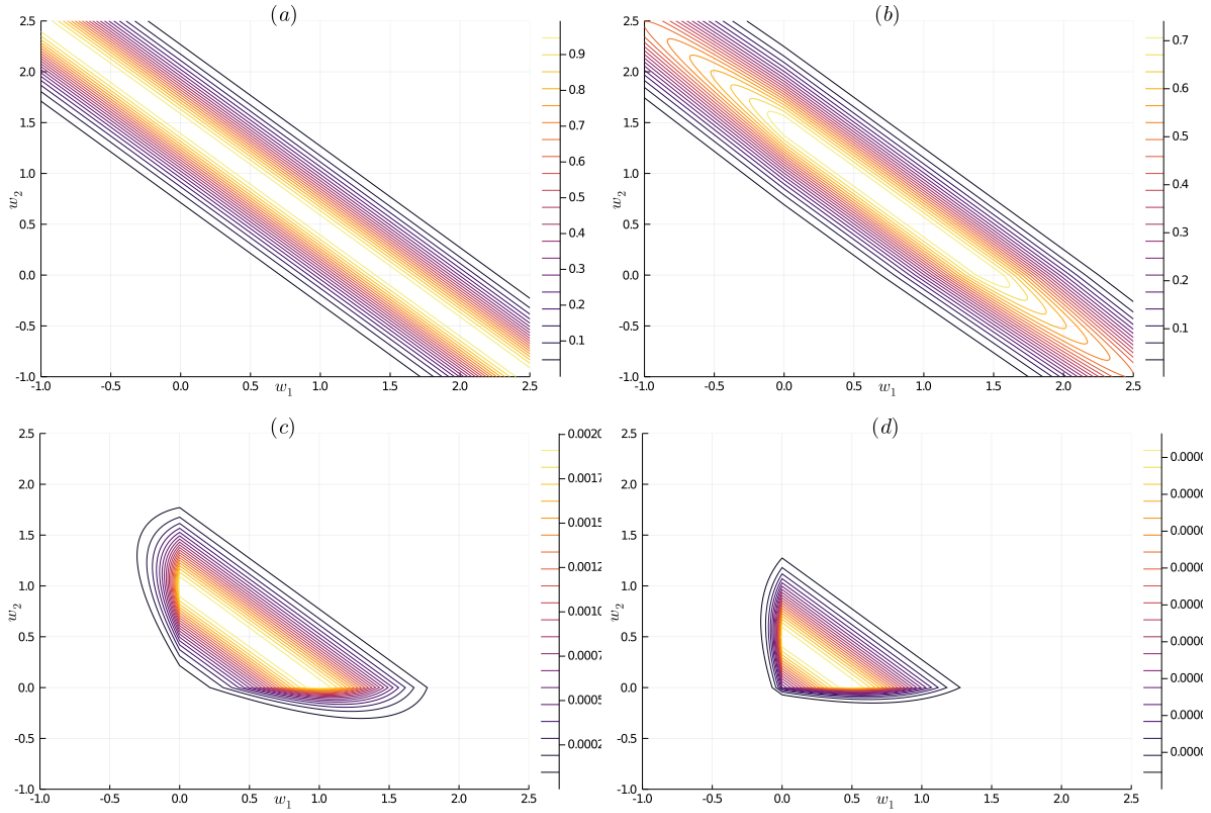
  kde ν značí počet stupňů volnosti.

[1] *Pierre Simon de Laplace* (1749–1827)

[2] *William Sealy Gosset* (1876–1937) – publikující pod pseudonymem *Student*



Obrázek 3.1: Contour plot přidání Spike & Slab k věrohodnostnímu členu lineární regrese, variance fixní $c^2 = 200$ a $\epsilon^2 = \frac{1}{200}$, (a) $\lambda = 0.9$, (b) $\lambda = 0.6$, (c) $\lambda = 0.3$, (d) $\lambda = 0.1$.



Obrázek 3.2: Contour plot přidání L_1 normy k věrohodnostnímu členu lineární regrese, (a) $\lambda = 0.002$, (b) $\lambda = 0.2$, (c) $\lambda = 5$, (d) $\lambda = 10$.

3.2 Řídkost a ARD

V sekci 2.1 a 2.2 jsme již setkali s parametrem α. Předpokládali jsme ho jako *skalár*. Nahraďme jej nyní vektorovým $\boldsymbol{\alpha}$ naznačeným v (3.7), který po složkách invertujeme a umístíme na diagonálu rozměrově odpovídající jednotkové matice $\mathbb{I}$, tj.

$$\operatorname{diag}\left(\boldsymbol{\alpha}^{-1}\right) = \operatorname{diag}\left(\alpha_0^{-1}, \alpha_1^{-1}, \ldots, \alpha_d^{-1}\right) = \begin{pmatrix} \alpha_0^{-1} & 0 & \ldots & 0 \\ 0 & \alpha_1^{-1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha_d^{-1} \end{pmatrix} \tag{3.8}$$

$$\boldsymbol{\alpha}^{-1}\mathbb{I} \overset{ozn.}{=} \operatorname{diag}(\boldsymbol{\alpha}^{-1}) \cdot \mathbb{I} \overset{3.8}{=} \begin{pmatrix} \alpha_0^{-1} & 0 & \ldots & 0 \\ 0 & \alpha_1^{-1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha_d^{-1} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix} = \begin{pmatrix} \alpha_0^{-1} & 0 & \ldots & 0 \\ 0 & \alpha_1^{-1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha_d^{-1} \end{pmatrix}. \tag{3.9}$$

Tím jsme vytvořili jistou kovarianční matici, jejíž prvky na diagonále (variance) náleží jednotlivým složkám vektoru parametrů $\boldsymbol{\theta}$. $\boldsymbol{\alpha}$ zde má význam přesnosti a platí:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\alpha}^{-1}\mathbb{I}) = \mathcal{N}(\mathbf{0}, \operatorname{diag}(\boldsymbol{\alpha}^{-1}) \cdot \mathbb{I}). \tag{3.10}$$

I zde můžeme klást požadavek na apriorní distribuce pro jednotlivé α_j. Připomeňme si vzorec (2.10):

$$p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X})p(\boldsymbol{\theta}|\alpha) = \exp\left(-\frac{1}{2}\|\mathbf{y} - \mathbb{X}\boldsymbol{\theta}\|_2^2 - \frac{1}{2}\alpha\|\boldsymbol{\theta}\|_2^2\right)$$

a rozšiřme o nové poznatky do více dimenzí v Euklidovském prostoru:

$$p(\mathbf{y}|\boldsymbol{\theta}, \mathbb{X})p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \exp\left(-\frac{1}{2}\|\mathbf{y} - \mathbb{X}\boldsymbol{\theta}\|_2^2 - \frac{1}{2}\left(\boldsymbol{\theta}^T \operatorname{diag}(\boldsymbol{\alpha})\boldsymbol{\theta}\right)\right) = \exp\left(-\frac{1}{2}\|\mathbf{y} - \mathbb{X}\boldsymbol{\theta}\|_2^2 - \frac{1}{2}\sum_j \alpha_j w_j^2\right). \quad (3.11)$$

Jelikož jsou komponenty $\boldsymbol{\alpha}$ nezávislé a zcela jistě platí

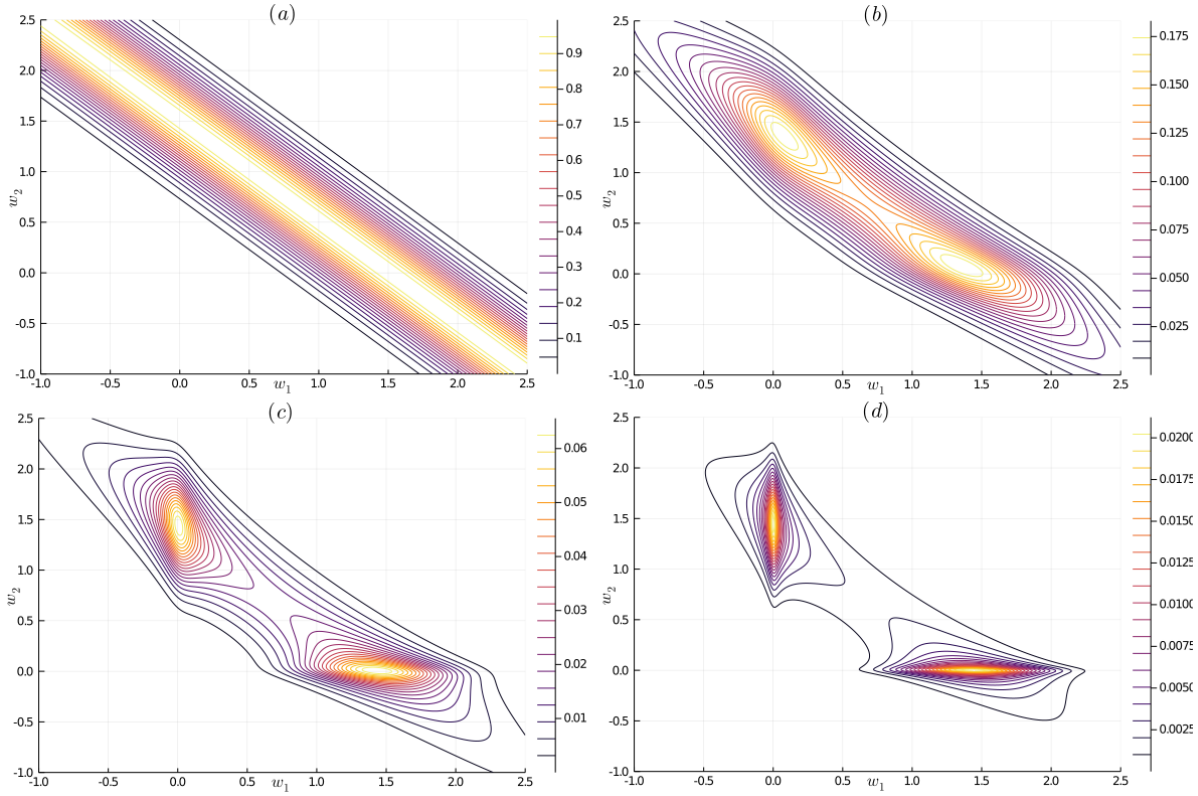
$$p(\boldsymbol{\alpha}) = \prod_j p(\alpha_j), \quad (3.12)$$

pak můžeme přiřadit každé komponentě α_j nějakou apriorní distribuci, která bude mít požadované vlastnosti na řídkost [19]. Pravděpodobnostní model z (2.10) přejde do tvaru

$$p(\mathbf{y}, \boldsymbol{\theta}|\mathbb{X}, \boldsymbol{\alpha}) = \mathcal{N}(\mathbb{X}\boldsymbol{\theta}, \mathbb{I})\mathcal{N}(\mathbf{0}, \boldsymbol{\alpha}^{-1}\mathbb{I}) \prod_j p(\alpha_j). \quad (3.13)$$

Za $p(\alpha_j)$ volíme například $\Gamma(0, 0)$ (viz Tabulka 2.1) nebo právě zobecněné Studentovo rozdělení s nulovou střední hodnotou (3.7).

Poznámka. Často se používá i symbol τ, který značí α^{-1}. Ve více dimenzích tedy $\boldsymbol{\tau} = \boldsymbol{\alpha}^{-1}$, kde invertování probíhá opět po složkách.



Obrázek 3.3: Contour plot přidání L_2 normy k věrohodnostnímu členu lineární regrese a apriorní distribuce pro $\alpha_j \sim \mathrm{St}(0, \sigma^2, \nu)$, (a) $\nu = 100$, $\sigma^2 = 1000$, (b) $\nu = 0.1$, $\sigma^2 = 1$, (c) $\nu = 0.01$, $\sigma^2 = 1$, (d) $\nu = 0.001$, $\sigma^2 = 1$.

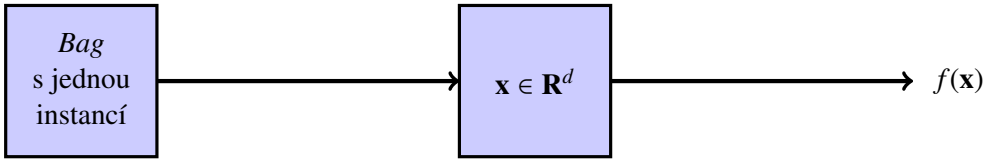
3.3 Více–instanční učení

Metoda více–instančního učení (MIL[3]) spadá do oblasti učení s učitelem. Jejím cílem je naučit se jistou *mapu* přiřazení mezi instancemi (*instances*) a jejich označeními (*labels*). Od klasického strojového učení se liší tím, že každý vzorek je popsán pomocí množiny instancí, příčemž velikost této množiny může nabývat jakéhokoli přirozeného čísla včetně nuly.

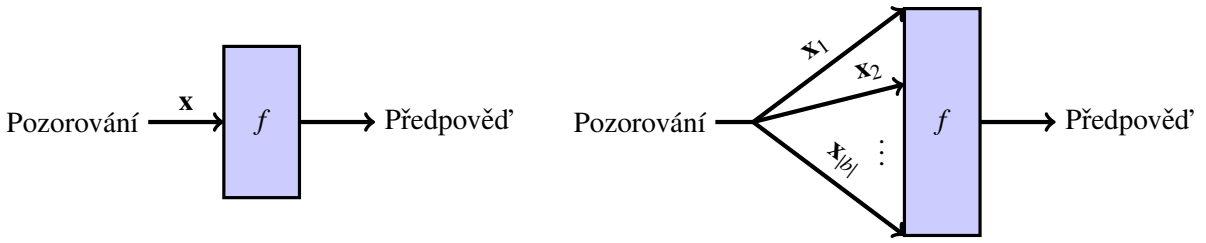
Zaveďme formální matematické značení [13]:

Definice 3.3.1. (Značení v MIL) Prostor všech instancí $\mathbf{x}_i$ označujeme $\mathscr{X}$. Prostor jejich označení jako $\mathscr{Y}$. Vzorek (*bag*) $b = \{\mathbf{x}_i \in \mathscr{X} | i \in \{1, \ldots, |b|\}\}$ ekvivalentně píšeme jako $b \in \mathscr{B} = \bigcup_{k>1} \left\{\mathbf{x}_i \in \mathscr{X} | i \in \hat{k}\right\}$. Parametrický prostor budeme značit $\mathscr{P}$.

Poznámka. Pokud $|b| = 1$, kde $|b|$ značí velikost *bagu*, pak úloha přechází na klasické strojové učení.



Obrázek 3.4: Klasické strojové učení neboli jedno–instanční (*single–instance learning*).



Obrázek 3.5: Rozdíl mezi klasickým a více–instančním strojovým učením (převzato z [9]).

Jeden vzorek (*sample*) ve více–instančním účení nazýváme *bag*. Jednotlivé *bagy* mohou obsahovat různé počty instancí (*instances*). Instance je též označována jako vektor příznaků a je popsána fixním počtem dimenzí. $\mathscr{B}$ nazýváme prostorem *bagů* a obsahuje všechny konečné podmnožiny $\mathscr{X}$. Každý z *bagů* má vlastní přiřazené označení (*label*). Ovšem nevíme, jakým způsobem jsou k tomuto označení přiřazeny instance v určitém *bagu*. Též ne každá instance je nezbytně důležitá. Můžeme mezi nimi najít takové, které nenesou žádnou informaci o jejich třídě, nebo zda jsou vůbec do nějaké zařaditelné.

Ve více–instančním učení definujeme modely jako:

$$f : \mathscr{B}(\mathscr{X}) \mapsto C, \tag{3.14}$$

kde f nazýváme klasifikátor a C označuje jistou konečnou třídu např. pro klasifikaci. Učení probíhá na dostupných datech v datasetu $D = \{(b_i, y_i) \in \mathscr{B} \times C | i \in \{1, \ldots, |D|\}\}$ [9].

[3] z anglického *Multiple–instance learning*

3.3.1 Paradigma vnořeného prostoru

Tento přístup přímo definuje vektorový prostor, ve kterém lze *bagy* reprezentovat a specifikovat tak vnoření každého *bagu* b do tohoto prostoru. Poté lze použít jakékoli techniky *ML* na vstupy o fixních dimenzích [12]. Ukážeme zde reprezentaci vnořeného prostoru pomocí neuronové sítě.

Nechť $\boldsymbol{\phi} : \mathscr{B} \mapsto \mathbf{R}^m$ je vektorové zobrazení, jehož každá složka je definována jako $\phi_i : \mathscr{B} \mapsto \mathbf{R}$ pro $\forall i \in \hat{m}$, tedy:

$$\boldsymbol{\phi}(b) = (\phi_1(b), \ldots, \phi_m(b)), \tag{3.15}$$

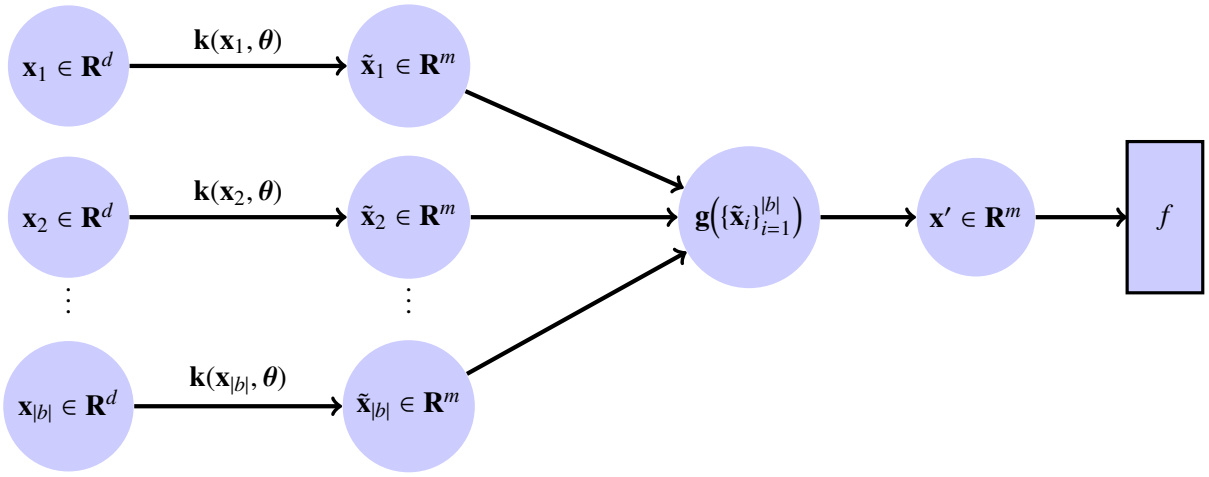
kde $\phi_i(b)$ lze dále definovat jako

$$\phi_i(b) = g\Big(\{k(\mathbf{x}, \theta_i)\}_{\mathbf{x} \in b}\Big). \tag{3.16}$$

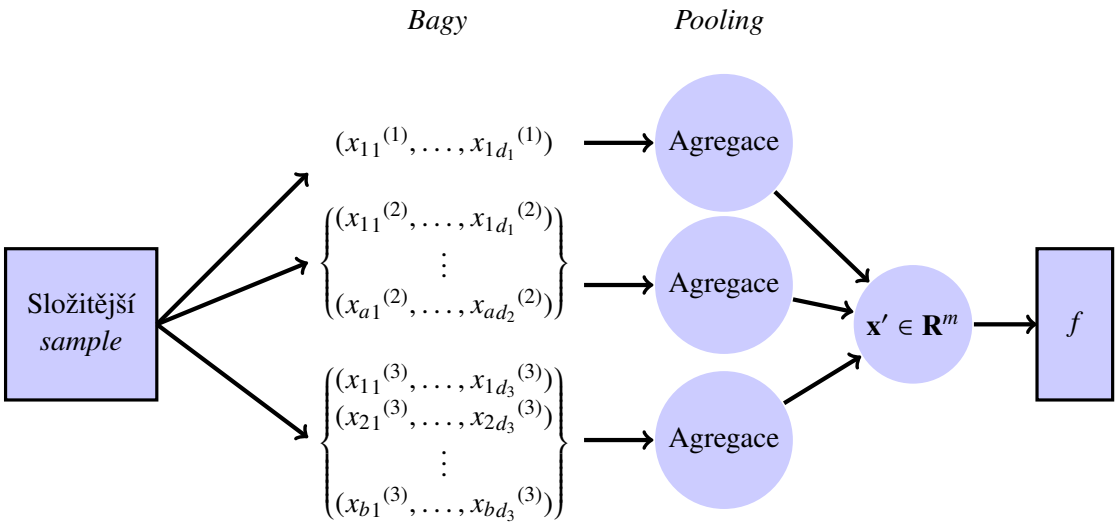
Zobrazení $k : \mathscr{X} \times \mathscr{P} \mapsto \mathbf{R}_0^+$ zde značí vhodnou *vzdálenostní funkci* parametrizovanou parametry $\boldsymbol{\theta}$ a $g : \cup_{n=1}^{+\infty} \mathbf{R}^l \mapsto \mathbf{R}$ nazýváme *pooling* funkcí [13].

Na model nastíněný v (3.15) a (3.16) lze pohlížet jako na neuronovou síť znázorněnou v Obrázku 3.6. Hlubší vrstva (či vrstvy) vytváří množinu *vzdálenostních funkcí* $\{k(\mathbf{x}, \theta_i)\}_{i=1}^m$, která nám zobrazí instance do prostoru $\mathbf{R}^m$. Následuje *pooling* vrstva, na jejímž výstupu bude jeden vektor o fixní dimenzi m. Poslední vrstva může sloužit např. jako klasifikátor. Díky této metodě lze zobrazit *bag* do prostoru o fixní dimenzi, se kterým umíme pracovat.

Velká výhoda takto definované reprezentace vnořeného prostoru pomocí neuronové sítě je v tom, že pokud zvolíme správnou *pooling* funkci $\mathbf{g}(.)$, pak všechny parametry funkcí $\mathbf{k}(.)$ mohou být optimalizovány standardním způsobem.



Obrázek 3.6: Nákres neuronové sítě optimalizující proces vnořování (převzato z [13]).



Obrázek 3.7: Jednoduchý příklad více–instančního učení, kde $x_{ij}{}^{(k)}$ znamená j–tou složku i–té instance v k–tém *bagu*.

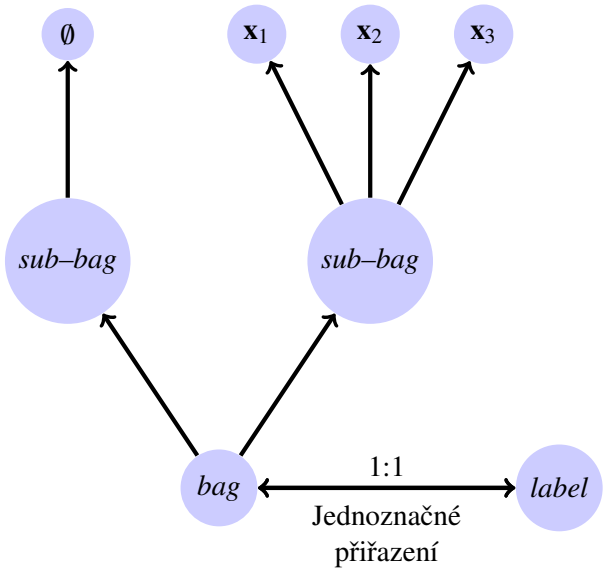
Na Obrázku 3.7 je vidět, že každý z *bagů* má vlastní agregační funkci (viz sekce 1.3). V tomto příkladě jsou uvedeny 3 *bagy*, kde nad každým z nich lze provést agregační operaci. Tím získáme tři vektory, které spojíme a získáme kompletní popis *samplu* ve vektorové podobě o fixní dimenzi, tj. $\mathbf{x}' \in \mathbf{R}^m$. To umožňuje vstup do klasifikátoru f.

3.3.2 Hierarchické MIL

Postupným vnořováním (*embeddingu*) *bagů* do určitých částí prostoru $\mathbf{R}^m$ vzniká tzv. vnořený prostor (*embedded space*). Každý výstup *embeddingu* může být vstupem do dalšího. Hovoříme tak o hierarchickém více–instančním učení (*HMill*).

HMill *sample* je tvořen uzly různých typů, které jsou uspořádány do podoby zakořeněného (orientovaného) stromu. Díky této myšlence lze data reprezentovat jako jejich hierarchii. Každý z listů obsahuje surová data a je možné k nim skrze strom najít přímou cestu. To je především obrovská výhoda tohoto přístupu.

V každé takové úrovni (uzlu) stromu budeme mít klasifikační modely, na které bude kladen požadavek řídkých parametrizací.



Obrázek 3.8: Příklad stromového popisu jednoduché hierarchie HMill. V jednotlivých instancí $\mathbf{x}_i$ jsou surová data.

Poznámka. Pro *bag* je *sub–bag* jeho potomkem. Na další úrovni stromu je *sub–bag* označen jako *bag*. Takto lze pokračovat libovolně dlouho. Jde pouze o přímočařejší terminologii.

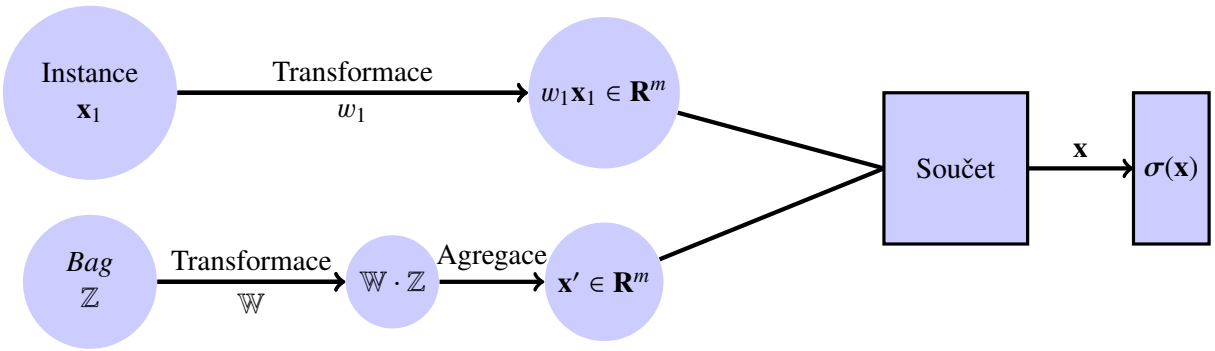
Příklad

Nechť je dána jednoduchá stromová struktua (3.17):

$$\mathbf{y} = \boldsymbol{\sigma}\left(w_1\mathbf{x}_1 + \max\left(\mathbb{W}\cdot\mathbb{Z}\right)\right), \tag{3.17}$$

kde $\mathbb{W}$ je parametrická transformační matice jednotlivých instancí seřazených v matici $\mathbb{Z}$. Maximum je zde agregačním operátorem nad tímto transformovaným *bagem*. Parametr w_1 zde představuje skalár a instance $\mathbf{x}_1$ je vektor s odpovídajícím počtem dimenzí jako výstup agregace.

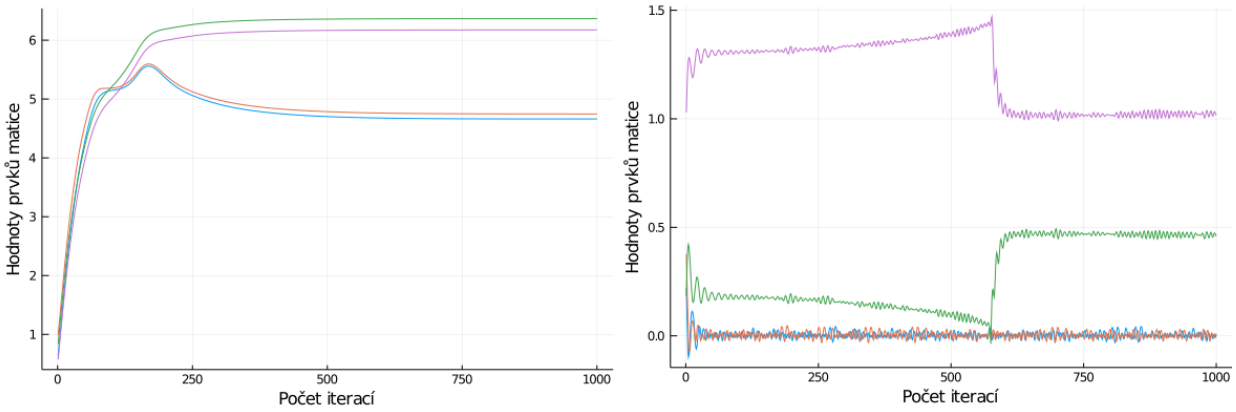
Rádi bychom se naučili prvky matice $\mathbb{W}$, která obsahuje celkem 4 parametry. Již umíme řešit klasické úlohy logistické regrese. Zde máme pouze navíc přidanou operaci agregace, čímž bychom mohli úlohu pojmenovat jako logistickou regresi se stromovou strukturou.



Obrázek 3.9: Grafické znázornění (3.17).

Generujme vektor $\mathbf{x}_1$ obsahující 1000 jednodimenzionálních dat z rozdělení $\mathcal{N}(0, 1)$ (tedy $m = 1000$) a nechť pravé $w'_1 = -3$ a $\mathbb{W}' = \begin{pmatrix} 2 & 3 \\ 0 & 0 \end{pmatrix}$ (rádi bychom řídkou prametrizaci). Velikost jednotlivých instancí v bagu $\mathbb{Z}$ budeme generovat pomocí Poissonova rozdělení se střední hodnotou 10.

Podobně jako v příkladě na konci druhé kapitoly se po agregační operaci jedná o stejnou úlohu se stejnou ztrátovou funkcí (2.21). Optimalizátor opět zvolíme metodu **ADAM** s krokem 0.01.



Obrázek 3.10: Proces učení parametrů matice $\mathbb{W}$. Vlevo bez penalizace. Vpravo s penalizací L_1 normy a koeficientem $\lambda = 0.01$.

Shoda dat s modelem je dobrá, ale parametry jsou po procesu učení úplně jiné než ty, které jsme generovali. Po penalizaci ztrátové funkce L_1 normou s koeficientem $\lambda = 0.01$ sice docílíme určitého řídkého řešení, ovšem ostatní parametry v matici nesouhlasí s původními. To znamená, že model lze vysvětlil jinak, než jsme chtěli. Optimalizace nám funguje, ale není dobře definovaná. Má příliš mnoho přijatelných řešení a my dostáváme jiné, než jsme původně požadovali.

Proto je třeba model rozšířit o reálná a robustnější data.

Závěr

Bakalářská práce byla svým zadáním zaměřena na klasifikaci dat popsaných stromovou strukturou. Aby bylo možné takto zadaný problém pochopit a řešit, bylo prvně třeba osvojit si obsáhlé spektrum partií matematiky v kombinaci s pojmy z oblasti optimalizace, strojového učení a umělé inteligence.

V úvodní kapitole byly zpracovány a vysloveny potřebné definice a teorémy ke snadnější orientaci v problematice. Též zde byla zavedena přehledná terminologie a symbolika, aby čtenář co nejlépe dokázal textu porozumět. Především byla podrobně zpracována teorie pravděpodobnosti, matematické statistiky a teorie grafů k vyslovení Bayesova teorému, odvození nástroje Elbo a pochopení konceptu stromů jakožto grafů. Byl zde nastíněn základní přístup k optimalizaci numerických úloh a jeho možného vylepšení pomocí adaptivních gradientních metod. Díky tomu mohl být ukázán dvojí přístup k řešení vybraných optimalizačních úloh. V neposlední řadě byly představeny neuronové sítě a jejich použití ve strojovém učení.

Ve druhé kapitole byly názorně aplikovány zavedené pojmy z úvodní kapitoly na jednoduché příklady, které poté mohly být numericky optimalizovány. Dále byla vyložena logistická regrese a její souvislost právě s neuronovými sítěmi. Práce se zabývá především binární klasifikací, ovšem na závěr kapitoly bylo nastíněno, jak lze přejít k vícetřídové klasifikaci pomocí speciální softmax funkce.

Závěrečná kapitola byla zaměřena především na základní vyložení vysvětlitelné umělé inteligenci a co vše s sebou tento pojem nese. Byla uvedena hlavní myšlenka, proč je mnohdy na modely nárokována řídká parametrizace a byly předvedeny různé přístupy a metody, jakými ji lze docílit, přičemž největší důraz byl kladen na metodu Automatic Relevance Determination s apriornem popsaným zobecněným Studentovým rozdělením. Dále bylo popsáno více–instanční učení, které zobecňuje klasické strojové učení na základě myšlenky popisu modelu vektory příznaků. Samotný závěr práce se věnoval konceptu vnořeného prostoru a jeho reprezentace pomocí neuronových sítí. Bakalářská práce byla zakončena náhledem do hierarchického více–instančního učení a jeho přínosu k řešení praktických úloh.

V navazující práci by bylo vhodné uchopit vybudovanou teorii demonstrovanou na jednodušších klasifikačních příkladech a aplikovat ji na složitější klasifikační úlohy reálných dat popsané obsáhlejšími stromovými strukturami za podmínky řídkých parametrizací. Veškeré numerické výpočty by byly opět prováděny v programovacím jazyce Julia.

Literatura

[1] BISHOP Ch. M. *Pattern recognition and machine learning*. New York: Springer, c2006. Information science and statistics. ISBN 978-0387-31073-2.

[2] ELTOFT T., KIM T a LEE T. *On the multivariate Laplace distribution*. IEEE Signal Processing Letters, 2006, 13.5: 300-303.

[3] EMMERT-STREIB F., YLI-HARJA O. a DEHMER M. *Explainable artificial intelligence and machine learning: A reality rooted perspective*. arXiv preprint arXiv:2001.09464, 2020.

[4] ISHWARAN H., KOGALUR U. B. a RAO J. S. *spikeslab: Prediction and Variable Selection Using Spike and Slab Regression*. R Journal, 2010, 2.2.

[5] JIROVSKÝ L. *Teorie grafů ve výuce na střední škole*. Praha, 2008. Diplomová práce. Univerzita Karlova.

[6] KINGMAN J. F. Ch. a TAYLOR S. J. *Introduction to Measure and Probability*. Cambridge University Press, 2008.

[7] KOVÁŘ J. a VAN DER MEER N.: *Zápisky z míry a pravděpodobnosti*. Učební text pro předmět Míra a pravděpodobnost. KM–FJFI–ČVUT v Praze, 2020.

[8] LOUIZOS, Ch., WELLING M. a KINGMA D. P. *Learning Sparse Neural Networks through L_0 Regularization.* arXiv preprint arXiv:1712.01312, 2017.

[9] MANDLÍK Š. *Mapping the Internet – Modelling Entity Interactions in Complex Heterogeneous Networks*. Prague, 2020. Master's Thesis. Czech Technical University in Prague.

[10] MOLNAR Ch. *Interpretable Machine Learning* [online]. [cit. 2020-06-27]. Dostupné z: `https://christophm.github.io/interpretable-ml-book/`

[11] NGUYEN T., RAICH R. a LAI P. *Jeffreys prior regularization for logistic regression*. In: 2016 IEEE Statistical Signal Processing Workshop (SSP). IEEE, 2016. p. 1-5.

[12] PEVNÝ T. a SOMOL P. *Discriminative models for multi-instance problems with tree structure.* In: Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. 2016. p. 83-91.

[13] PEVNÝ T. a SOMOL P. *Using neural network formalism to solve multiple-instance problems.* In: International Symposium on Neural Networks. Springer, Cham, 2017. p. 135-142.

[14] PIIRONEN J. a VEHTARI A. *Sparsity information and regularization in the horseshoe and other shrinkage priors.* Electronic Journal of Statistics, 2017, 11.2: 5018-5051.

[15] PUDIL P., NOVOVIČOVÁ J. a KITTLER J. *Floating search methods in feature selection.* Pattern recognition letters, 1994, 15.11: 1119-1125.

[16] ROBERT Ch. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007. ISBN 9780387715988.

[17] SCHMIDT M. *CPSC 540: Machine Learning, Stochastic Subgradient* [online]. [cit. 2020-06-10]. Dostupné: https://www.cs.ubc.ca/~schmidtm/Courses/540-W19/L10.pdf

[18] SCHMIDT M. *Least squares optimization with L1-norm regularization*. CS542B Project Report, 2005, 504: 195-221.

[19] ŠMÍDL V. *Linear Regression, Automatic Relevance Determination* [online]. [cit. 2020-06-10]. Dostupné z: http://staff.utia.cas.cz/smidl/files/hbm2020/prezentace03_20.pdf

[20] TRUDEAU R. J. *Introduction to graph theory.* Courier Corporation, 2013. ISBN 9781684112319.

[21] VESELOVSKÝ M. *Plánování cesty robotů pomocí posilovaného učení*. Brno, 2013. Diplomová práce. Vysoké učení technické v Brně.

[22] WIPF P. D. a NAGARAJAN S. S. *A new view of automatic relevance determination.* In: Advances in neural information processing systems. 2008. p. 1625-1632.

[23] YANG X. *Understanding the variational lower bound* [online]. [cit. 2019-12-10]. Dostupné z: http://users.umiacs.umd.edu/~xyang35/files/understanding-variational-lower.pdf