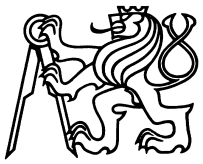
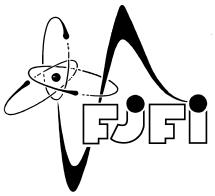


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



Generativní modely dat popsaných stromovou strukturou

Generative models of tree structured data

Bakalářská práce

Autor: **Jakub Bureš**

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

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

Akademický rok: 2019/2020

1. 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.

2. Seznamte se se základními generativními modely dat popsaných vektorem příznaků. Zvláštní pozornost věnujte metodám typu autoencoder a jejich variační formě. Demonstrujte vlastnosti modelů na jednoduchých příkladech. V maximální míře využijte dostupné knihovny pro generativní modely.

3. Navrhněte několik příkladů typů dat se stromovou strukturou a pro každý z nich navrhněte generativní model. Navrhněte algoritmus pro určení jeho parametrů z dat a diskutujte vhodnost jednotlivých architektur neuronových sítí.

4. Seznamte se s různými druhy apriorních rozložení používaných na latentní proměnné autoencoderu. Odvoďte algoritmy odhadu jejich parametrů a srovnejte jejich výsledky se základním modelem. Diskutujte výsledné odhady.

5. Vyvinutou metodu aplikujte na vhodně zvolená reálná data a diskutujte vliv zvoleného apriorního rozložení na výsledky.

Poděkování:
Chtěl bych zde poděkovat především svému školiteli panu doc. Ing. Václavu Šmídlovi, Ph.D. za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé bakalářské práce. Bylo z jeho strany potřeba i mnoho trpělivosti a mnohdy i neúměrné množství času, neboť toto téma pro mě bylo zcela nové a ze začátku velmi náročné.

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

V Praze dne 22. července 2020 Jakub Bureš

Název práce:

Generativní modely dat popsaných stromovou strukturou

Autor: Jakub Bureš

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, 182 00 Praha 8,

Konzultant: doc. Ing. Tomáš Pevný, Ph.D.
Katedra počítačů, FEL ČVUT Praha, Technická 1902/2, 166 27 Praha 6 - Dejvice

Abstrakt: Tato bakalářská práce se zabývá generativními modely a jejich možným využitím v internetové bezpečnosti. Hlavním cílem je hledat takové distribuce, která jsou schopna generovat jednoduché stromové struktury. Seznámíme se s moderními metodami a postupy na bázi umělé inteligence a na jednoduchých příkladech je aplikujeme. To zahrnuje neuronové sítě, koncept multi–instančního učení, vnořeného prostoru a metody variačního autoencoderu. V první části vhodně rozebereme nutný matematický aparát, ve druhé části se budeme věnovat generativním modelům a v poslední třetí části stromovým strukturám.

Klíčová slova: distribuce, generativní modely, multi–instační učení, stromové struktury,

Title:

Generative models of tree structured data

Author: Jakub Bureš

Abstract: This bachelor's thesis deals with generative models and their possible utilization in the internet safety. The main goal of this thesis is to find such distributions, which are capable of generating simple tree structures. We get to know modern methods and approaches using artificial intelligence and we apply them on simple examples. Those methods includes neural networks, concept of the multi-instance learning, the embedded-space and methods of variational autoencoders. In the first part we look into necessary calculus, in the second part we dive into generative models and in the last third part, we look into tree structures.

Key words: distribution, generative models, multi–instance learning, tree structures

Obsah

Úvod

Internet se stal nedílnou součástí našich životů a mnozí si už ani nedokáží představit, jak by bez něho řešili každodenní starosti. Se zvyšujícím se provozem v internetu, se zvyšuje též počet pokusů o jeho zneužití, ať už pomocí virů, odposlouchávání, botnetů či malwaru obecně. Tradiční obranná řešení se spoléhají na identifikaci předem stanovených znaků, kterými se malware liší od neškodného programu. Inteligence a adaptivita malwaru však neustále roste a prakticky tak znemožňuje nalezení deterministických pravidel či postupů k jeho detekci.

Moderní metody detekce využívají umělou inteligenci (AI), zejména pak neuronové sítě a sestavují diskriminativní modely, čímž se snaží klasifikovat počítač do třídy nakažený nebo nenakažený a to na základě jejich HTTP(S) [4], [15]. Tyto metody na rozdíl od klasického strojového učení, využívají tzv. multi-instanční učení, kde jsou data hierarchicky rozložena do stromových struktur, přesněji řečeno do tzv. vektorů příznaků. Mnoho autorů se pokusilo tento koncept vylepšit, přičemž nejnovějším fenoménem je unifikovaná knihovna HMill [2].

My k tomuto problému v této bakalářské práci přistoupíme jinak a zaměříme se na modely generativní, přičemž ještě nebudeme řešit reálné problémy. Práce je koncipována do třech kapitol v logickém sledu. Na úrovni této práce se budeme zabývat i nezbytným matematickým aparátem, který bude shrnut v první kapitole. Hlavním účelem této kapitoly bude odvodit ELBO (Evidence Lower Bound). Ve druhé části budeme diskutovat rozdíly mezi generativními a diskriminativními modely, definujeme neuronovou síť a představíme metodu variační autoencoderu. V poslední třetí části se budeme věnovat stromovým strukturám, představíme multi-instanční učení a jeho rozdíl oproti klasickému strojovému učení a nakonec koncept vnořeného prostoru. Problematiku každé kapitoly se vždy pokusíme osvětlit pomocí jednoduchých příkladů. Při jejich řešení budeme výhradně používat programovací jazyk Julia [16]. Primárním cílem celé práce je pak najít takovou distribuci, která umí generovat stromové struktury. Tato práce by měla sloužit jako odrazový můstek pro hlubší pochopení této problematiky.

Kapitola 1

Teorie

1.1 Optimalizace

Optimalizace je matematická úloha, jejíž snahou je nalezení takových hodnot proměnných, pro které daná funkce nabývá minima či maxima. My se budeme snažit najít minimální hodnoty vektoru parametrů θ tzv. ztrátové funkce *(loss function)*. Ztrátovou funkci budeme dle anglického výrazu značit $L(\theta)$. Minimalizací ztrátové funkce získáme odhad parametrů

$$\hat{\theta} = \arg\min_{\theta} L(\theta), \tag{1.1}$$

který budeme vždy značit pomocí stříšky. Existuje nespočet způsobů jak danou funkci minimalizovat. My budeme výhradně používat metodu zvanou Gradient Descent.

1.1.1 Gradient Descent

Jedná se o iterativní optimalizační metodu. Minimalizujeme $L(\theta)$, tedy derivujeme dle vektoru parametrů θ, díky čemuž dostaneme $\nabla_\theta L(\theta)$. Symbol ∇_θ značí gradient funkce $L(\theta)$ přes všechny hodnoty θ. Použijeme bod θ_0 funkce $L(\theta)$ jako výchozí bod. Jelikož gradient udává směr nejvyššího růstu, pohybujeme se ve směru záporného gradientu a to s krokem $h \in \mathbb{R}_+$. Matematickou interpretaci tohoto postupu můžeme vyjádřit následujícím zápisem

$$\theta_{n+1} = \theta_n - h \cdot \nabla_\theta L(\theta_n). \tag{1.2}$$

Tento postup provádíme, dokud se nenacházíme v minimu funkce, čímž získáme vektor parametrů $\hat{\theta}$, jak je popsáno v (1.1). Ačkoliv se tento algoritmus může zdát na první pohled jako silný nástroj při řešení optimalizačních úloh, ve skutečnosti je opak pravdou. Ve velkých dimenzích a obrovském množství dat je tato metoda pomalá a prakticky se nepoužívá.

ADAM

Kvůli důvodům uvedeným výše, používáme vylepšenou variantu metody Gradient Descent, jedná se o tzv. Stochastic Gradient Descent (dále jen SGD). V této rodině existuje několik algoritmů výpočtu extrému. My budeme využívat adaptivní iterační metodu ADAM *(Adaptive*

Moment Estimation) [6], která navíc používá druhý moment gradientu. Zatímco klasický Gradient Descent má krok stále stejný, u metody ADAM je krok h adaptivní. Více ji specifikovat v tomto textu nebudeme. Pro nás je důležité, že je tento algoritmus silnější a mnohem rychlejší v řešení složitých optimalizačních úloh. Další používané algoritmy z SGD jsou RMSprop, Adagrad nebo AdaMax.

1.1.2 Metoda nejmenších čtverců

Metoda nejmenších čtverců [13] je nejzákladnější metoda pro hledání nejlepšího proložení určitých dat nějakou křivkou. Představíme zjednodušenou alternativu, jak tuto metodu odvodit. Předpokládejme, že máme množinu $\mathbf{x} = \{x_i\}_{i=1}^n$, kde ke každému x_i máme právě jedno pozorování y_i, které je zatíženo nějakou neznámou chybou ε_i. Označme $\mathbf{y} = \{y_i\}_{i=1}^n$, komplexně zapsáno zobrazením jako $(x_1, \ldots, x_n) \longmapsto (y_1, \ldots, y_n)$. Naším cílem je najít nejlepší proložení dat, čili fit, pomocí polynomické funkce řádu $p \leq n$ a to ve tvaru

$$\hat{y}(x, \theta) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_p x^p = \sum_{i=0}^{p} \theta_i x^i, \tag{1.3}$$

která je lineární v neznámých parametrech $\theta = \left(\theta_0, \theta_1, \ldots, \theta_p\right)^\mathsf{T}$. Takové modely nazýváme lineární. Jelikož se jedná o tak jednoduché modely, jejich míra využití je značně omezena. O tom jak tyto modely vylepšit, se dozvíme v kapitole 2.2.
Abychom našli ten nejlepší možný fit, je nutno minimalizovat ztrátovou funkci, která má v tomto případě tvar

$$L(\theta) = \sum_{i=1}^{n} \left[y_i - \hat{y}(x_i, \theta)\right]^2 = \sum_{i=1}^{n} \varepsilon_i^2 = (\mathbf{y} - \mathbb{X} \cdot \theta)^\mathsf{T} (\mathbf{y} - \mathbb{X} \cdot \theta). \tag{1.4}$$

Tato funkce znázorňuje čtverec vzdálenosti pozorovaní $\mathbf{y}$ k hledané funkci $\hat{y}(x, \theta)$, jenž chceme mít co nejmenší - proto metoda nejmenších čtverců. Matice $\mathbb{X}$ je tvaru

$$\mathbb{X} = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^p \\ 1 & x_2 & x_2^2 & \ldots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^p \end{pmatrix}. \tag{1.5}$$

Odhadovat parametry θ můžeme numericky a to pomocí gradientní metody. Využijeme pravidla pro výpočet derivace dle vektorů [10] a najdeme tak gradient ztrátové funkce

$$\nabla_\theta L(\theta) = 2\mathbb{X}^\mathsf{T}(\mathbb{X} \cdot \theta - \mathbf{y}). \tag{1.6}$$

Dále postupujeme pomocí rovnice (1.2), dokud nezískáme

$$\hat{\theta} = \arg\min_{\theta} L(\theta). \tag{1.7}$$

Toto ovšem není jediný způsob odhadu parametrů. Metoda nejmenších čtverců má i analytické řešení. Systém rovnic můžeme zapsat následující formou

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & \dots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^p \end{pmatrix} \cdot \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_p \end{pmatrix}.$$

Pro jednoduchost budeme tímto zápisem rozumět následující rovnici

$$\mathbf{y} = \mathbb{X} \cdot \theta + \epsilon. \tag{1.8}$$

Naším cílem je opět získání odhadu parametrů θ. Jelikož bude chyba ϵ při $\hat{\theta}$ nulová, můžeme předchozí rovnici přepsat následovně

$$\mathbf{y} = \mathbb{X} \cdot \hat{\theta}. \tag{1.9}$$

Nyní obě strany rovnice vynásobíme zleva výrazem $\mathbb{X}^\mathsf{T}$. Tím nám rovnice přejde do tvaru

$$\mathbb{X}^\mathsf{T} \cdot \mathbf{y} = \mathbb{X}^\mathsf{T} \cdot \mathbb{X} \cdot \hat{\theta}.$$

Teď už stačí rovnici zleva vynásobit inverzní maticí $(\mathbb{X}^\mathsf{T} \cdot \mathbb{X})^{-1}$. Dostaneme tak konečné řešení odhadu parametrů

$$\hat{\theta} = (\mathbb{X}^\mathsf{T}\mathbb{X})^{-1}\mathbb{X}^\mathsf{T}\mathbf{y}. \tag{1.10}$$

Tento postup zahrnuje i lineární regresi pro hodnotu $p = 1$, tzn. že bychom hledali funkci ve tvaru $\hat{y}(x, \theta) = \theta_0 + \theta_1 x$. Matice $\mathbb{X}$ by tak obsahovala pouze první dva sloupce.

1.2 Úvod do pravděpodobnosti a Bayesovská statistika

K hledání pravděpodobnostního modelu je potřeba znát pravděpodobnostní počet [8] a statistiku [9] . Uvedeme zde nezbytné znalosti a ucelíme značení.

1.2.1 Pravděpodobnostní míra

Definice 1.2.1 (Kolmogorova definice pravděpodobnosti)**.** Mějme neprázdnou množinu Ω vybavenou σ-algebrou $\mathscr{A}$, tedy souborem podmnožin obsahujícím Ω a uzavřeným na doplňky a spočetná sjednocení. Pak libovolnou funkci $P : \mathscr{A} \to \mathbb{R}$, která splňuje

1. $(\forall A \in \mathscr{A})(P(A) \geq 0)$,
2. $P(\Omega) = 1$,
3. $\forall A_j$ disjunktní platí $P(\sum_{j=1}^{\infty} A_j) = \sum_{j=1}^{\infty} P(A_j)$,

nazýváme pravděpodobnostní mírou.

Věta 1.2.1 (Vlastnosti P)**.** *Mějme pravděpodobnostní prostor* $(\Omega, \mathscr{A}, P)$ *a nechť* $(\forall j \in \mathbb{N})(A_j \in \mathscr{A})$ *a* $B \in \mathscr{A}$*. Pak platí*

1. $P(\emptyset) = 0$,
2. *Aditivita:* $P(\sum_{j=1}^{n} A_j) = \sum_{j=1}^{n} P(A_j)$,
3. *Monotonie:* $A \subset B \Rightarrow P(A) \leq P(B)$,
4. *Subtraktivita:* $A \subset B \Rightarrow P(B \setminus A) = P(B) - P(A)$,
5. *Omezenost:* $(\forall A \in \mathscr{A})(P(A) \leqslant 1)$,
6. *Komplementarita:* $A \in \mathscr{A} \Rightarrow P(A^C) = 1 - P(A)$.

Definice 1.2.2 (Podmíněná pravděpodobnost)**.** Nechť $A, B \in \mathscr{A}$ a $P(B) > 0$. Pak definujeme podmíněnou pravděpodobnost vztahem

$$P(A|B) = \frac{P(A, B)}{P(B)}. \tag{1.11}$$

Věta 1.2.2 (Součinové pravidlo)**.** *Nechť* $A_1, \ldots, A_n \in \mathscr{A}$ *a dále nechť také* $P(A_1, \ldots, A_n) > 0$. *Potom platí*

$$P(A_1, \ldots, A_n) = P(A_1) \cdot P(A_2|A_1) \cdot P(A_3|A_2, A_1) \cdot \ldots \cdot P(A_n|A_1, \ldots, A_{n-1}). \tag{1.12}$$

Věta 1.2.3 (Bayesova věta)**.** *Nechť* $A \in \mathscr{A}$ *a* $P(B) \neq 0$. *Potom platí*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{1.13}$$

$P(A)$ nazýváme prior, $P(A|B)$ posterior a jmenovatel $P(B)$ je často nazýván jako evidence.

Věta 1.2.4 (Nezávislost jevů)**.** *Nechť* $A_j \in \mathscr{A}(\forall j \in \mathbb{N})$. *Potom jevy nazveme nezávislé právě tehdy, když platí podmínka*

$$P(A_1, \ldots, A_k) = \prod_{i=1}^{k} P(A_i). \tag{1.14}$$

1.2.2 Hustoty pravděpodobnosti

Primárním cílem generativního modelování je hledání hustoty pravděpodobnosti (pravděpodobnostního rozdělení, distribuce) daných dat. Výhodou je, že pro hustotu pravděpodobnosti můžeme využívat stejným způsobem pravidlo podmíněnosti (1.11), tak i součinové pravidlo (1.12) a Bayeseovo pravidlo (1.13). Toto se pro nás ukáže jako naprosto klíčové.

Definice 1.2.3 (Náhodná veličina)**.** Máme prostor $(\Omega, \mathscr{A})$, potom funkci $\mathbf{X} = (X_1, \ldots, X_n) : (\Omega, \mathscr{A}) \mapsto (\mathbb{R}^n, \mathscr{B}_n)$, kde $\mathscr{B}_n$ značí borelovskou σ-algrebru v $\mathbb{R}^n$, nazveme náhodnou veličinou.

Definice náhodné veličiny vypadá poněkud složitě. Pro nás je důležité, že veškerá **pozorovaná data** jsou náhodnou veličinou. Každý datový záznam bude většinou nezávislý a stejně rozdělený, což budeme značit zkratkou i.i.d. *(Independently Identically Distributed)*.

Definice 1.2.4 (Hustota pravděpodobnosti)**.** Hustotou pravděpodobnosti náhodné veličiny $\mathbf{X}$ rozumíme spojitou funkci $p_{\mathbf{X}}(\mathbf{x})$, která splňuje následující dvě podmínky

1. $\forall \mathbf{x}$, $p_{\mathbf{X}}(\mathbf{x}) \geq 0$,
2. $\int\limits_{\mathbb{R}^n} p_{\mathbf{X}}(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$.

Obdobou hustoty pravděpodobnosti pro diskrétní náhodnou veličinu je **pravděpodobnostní funkce** $P[\mathbf{X} = \mathbf{x}]$ splňující

1. $\forall \mathbf{x}$, $P[\mathbf{X} = \mathbf{x}] \geq 0$,
2. $\sum_{\mathbf{x}} P[\mathbf{X} = \mathbf{x}] = 1$.

My se většinou omezíme na jednorozměrné a spojité náhodné veličiny. V takovém případě budeme psát X a $p_X(x)$. V případě, že bude mít náhodná veličina nějakou hustotu pravděpodobnosti, budeme to zapisovat pomocí $\sim$, tedy $X \sim p_X(x)$. Index budeme vynechávat, protože bude jasné, ke které náhodné veličině hustota patří. Podívejme se nyní, jak určit hustotu transformované náhodné veličiny.

Poznámka. *V této práci se můžeme setkat s termíny hustota, rozdělení nebo distribuce – všechny tyto termíny budou pro jednoduchost ekvivalentní.*

Věta 1.2.5 (Transformace náhodné veličiny)**.** *Nechť* $\boldsymbol{X} \sim p_{\boldsymbol{X}}(\boldsymbol{x})$ *a nechť* $h : \mathbb{R}^n \mapsto \mathbb{R}^n$ *je regulární a prosté zobrazení na množině* H*, takové že* $\int_H p_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 1$*. Potom je* $\boldsymbol{Y} = h(\boldsymbol{X})$ *náhodná veličina a její hustota* $\forall \boldsymbol{y} \in h(H)$ *má následující tvar*

$$p_{\boldsymbol{Y}}(\boldsymbol{y}) = p_{\boldsymbol{X}}\left(h^{-1}(\boldsymbol{y})\right) \cdot |\det \mathbb{J}_{h^{-1}}(\boldsymbol{y})|\,. \tag{1.15}$$

Symbol $\det \mathbb{J}_{h^{-1}}$ *zde značí determinant z Jacobiho matice inverzního zobrazení k* h*.*

Nyní ukážeme, jak určit vybrané charakteristiky náhodné veličiny. Bude se jednat o střední hodnotu, rozptyl a entropii.

Definice 1.2.5 (Střední hodnota náhodné veličiny)**.** Má–li náhodná veličina $\mathbf{X} \in \mathscr{L}_1$ spojitou hustotu pravděpodobnosti $p(\mathbf{x})$, definujeme její střední (očekávanou) hodnotu $\mathbb{E}[\mathbf{X}]$, alternativně značeno $\langle \mathbf{X} \rangle$, vztahem

$$\mathbb{E}[\mathbf{X}] = \int\limits_{\Omega} \mathbf{X}\mathrm{d}P = \int\limits_{\mathbb{R}^n} \mathbf{x} \cdot p(\mathbf{x})\mathrm{d}\mathbf{x}. \tag{1.16}$$

Pro diskrétní náhodnou veličinu s pravděpodobnostní funkcí $P[\mathbf{X} = \mathbf{x}]$ platí

$$\mathbb{E}[\mathbf{X}] = \sum_k \mathbf{x}_k \cdot P[\mathbf{X} = \mathbf{x}_k]\,. \tag{1.17}$$

Definice 1.2.6 (Rozptyl náhodné veličiny)**.** Má–li náhodná veličina $X \in \mathscr{L}_2$ spojitou hustotu pravděpodobnosti $p(x)$, definujeme rozptyl (varianci) $\mathbb{D}[X]$, alternativně značeno $\mathrm{Var}(X)$, vztahem

$$\mathbb{D}[X] = \mathbb{E}[X - \mathbb{E}[X]]^2 = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 . \tag{1.18}$$

Pro vícerozměrnou náhodnou veličinu $\mathbf{X} \in \mathscr{L}_2$ je variance $n \times n$ rozměrná matice a nazýváme ji kovarianční. Je definována vztahem

$$\mathbb{D}[\mathbf{X}] = \mathbb{E}\left[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^{\mathsf{T}}\right]. \tag{1.19}$$

Definice 1.2.7 (Entropie)**.** Má–li náhodná veličina $\mathbf{X} \in \mathscr{L}_1$ spojitou hustotu pravděpodobnosti $p(\mathbf{x})$, definujeme entropii náhodné veličiny $\mathbb{H}[\mathbf{X}]$, vztahem

$$\mathbb{H}[\mathbf{X}] = \mathbb{E}[-\log p(\mathbf{x})], \tag{1.20}$$

kde log značí přirozený logaritmus. Stejně jako pro střední hodnotu, můžeme entropii definovat pro diskrétní náhodnou veličinu

$$\mathbb{H}[\mathbf{X}] = -\sum_k P[\mathbf{X} = \mathbf{x}_k] \cdot \log P[\mathbf{X} = \mathbf{x}_k] . \tag{1.21}$$

Poznámka. *Kvůli zjednodušení zápisu nebudeme později uvádět integrační množinu – pokud není uvedeno jinak, budeme předpokládat, že se integruje přes celý nosič hustoty.*

V dalším textu uvedeme příklady spojitých či diskrétních rozdělení a pro přehlednost jejich výše zmíněné charakteristiky, jelikož je v této práci budeme využívat.

Poissonovo rozdělení

Poissonovo rozdělení popisuje diskrétní náhodnou veličinu. Většinou se jedná o počet výskytu určitého jevu v daném intervalu. Důležité je, že tyto jevy nastávají nezávisle na sobě. Pravděpodobnostní funkci Poissonova rozdělení vyjadřujeme pomocí parametru λ ve tvaru

$$\mathrm{Po}(\lambda) = \frac{\lambda^k}{k!}\mathrm{e}^{-\lambda}. \tag{1.22}$$

- $\mathbb{E}[X] = \lambda$
- $\mathbb{D}[X] = \lambda$
- $\mathbb{H}[X] = \lambda(1 - \log(\lambda)) + \mathrm{e}^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k \log(k!)}{k!}$

Rovnoměrné rozdělení

Je jedním z nejjednodušších rozdělení pro spojité proměnné. Rovnoměrné rozdělení, někdy také nazýváno uniformní, přiřazuje všem hodnotám stejnou pravděpodobnost. Je definováno na intervalu (a, b) a jeho hustotu můžeme vyjádřit následující formou

$$\mathrm{U}(a, b) = \begin{cases} \frac{1}{b-a}, & \text{pro } x \in (a, b) \\ 0, & \text{jinak} \end{cases} . \tag{1.23}$$

- $\mathbb{E}[X] = \frac{1}{2}(a+b)$
- $\mathbb{D}[X] = \frac{1}{12}(b-a)^2$
- $\mathbb{H}[X] = \log(b-a)$

Normální rozdělení

Nejdůležitější hustota pravděpodobnosti pro spojité proměnné se nazývá Normální nebo také Gaussovo rozdělení. Jeho hustota je definována $\forall x \in \mathbb{R}$ pomocí dvou parametrů $\mu \in \mathbb{R}$ a $\sigma^2 > 0$ vztahem

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}. \tag{1.24}$$

- $\mathbb{E}[X] = \mu$
- $\mathbb{D}[X] = \sigma^2$
- $\mathbb{H}[X] = \frac{1}{2}\log\left(2\pi e\sigma^2\right)$

Budeme využívat i n-rozměrnou variantu Gaussova rozdělení, jehož hustota je definováno vztahem

$$\mathcal{N}(\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}, \tag{1.25}$$

kde Σ je kovarianční matice a $\boldsymbol{\mu}$ je vektor středních hodnot.

- $\mathbb{E}[X] = \boldsymbol{\mu}$
- $\mathbb{D}[X] = \Sigma$
- $\mathbb{H}[X] = \frac{1}{2}\log\det(2\pi e\Sigma)$

Gamma rozdělení

Gamma rozdělení je definováno stejně jako Normální rozdělení pomocí dvou parametrů $\alpha > 0$ a $\beta > 0$. Jeho hustota pravděpodobnosti má smysl pro $\forall x > 0$ a můžeme ji najít v několika možných tvarech. My uvedeme tento

$$\text{Gamma}(\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp\{-\beta x\}, \tag{1.26}$$

kde $\Gamma(\alpha)$ značí gamma funkci. Stejně jako u předchozích rozdělení uvedeme některé důležité charakteristiky.

- $\mathbb{E}[X] = \frac{\alpha}{\beta}$
- $\mathbb{D}[X] = \frac{\alpha}{\beta^2}$
- $\mathbb{H}[X] = \alpha - \log\beta + \log\Gamma(\alpha) + (1-\alpha)\psi(\alpha)$

Přičemž funkce $\psi(\alpha)$ značí digamma funkci, čili $\psi(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$.

Inverzní gamma rozdělení

Inverzní gamma rozdělení je velmi podobné gamma rozdělení akorát pro převrácenou hodnotu x, je tedy opět popsáno dvěma parametry $\alpha > 0$ a $\beta > 0$ a definováno pro $\forall x > 0$. Jeho hustotu můžeme zapsat následovně

$$\text{invGamma}(\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left\{-\frac{\beta}{x}\right\} \tag{1.27}$$

Střední hodnota a rozptyl invGamma(α,β) nejsou ale definovány pro $\alpha > 0$.

- $\mathbb{E}[X] = \frac{\beta}{\alpha-1}$, pro $\alpha > 1$
- $\mathbb{D}[X] = \frac{\beta^2}{(\alpha-1)^2(\alpha-2)^2}$, pro $\alpha > 2$
- $\mathbb{H}[X] = \alpha + \log\beta + \log\Gamma(\alpha) - (1+\alpha)\psi(\alpha)$

Poznámka. *V textu budeme používat výraz* invGamma$(0, 0+)$*, kde symbol* $0+$ *značí číslo velmi blízké* 0*. Budeme tím rozumět hustotu ve tvaru*

$$\text{invGamma}(0, 0+) = \frac{1}{x}.$$

1.2.3 Bayesovská metoda nejmenších čtverců

Uvažujme standardní problém na nejmenší čtverce (1.8), tzn.

$$\mathbf{y} = \mathbb{X} \cdot \theta + \epsilon,$$

předpokládáme však, že pro jednu složku vektoru ϵ platí $\varepsilon_i \sim \mathcal{N}\left(0, \sigma^2\right)$ a navíc jsou tyto složky i.i.d. Díky vlastnostem Gaussova rozdělení můžeme určit hustotu

$$p(\mathbf{y}|\mathbb{X}) = \mathcal{N}\left(\mathbb{X} \cdot \hat{\theta}, \sigma^2 \cdot \mathbb{I}\right), \tag{1.28}$$

kde $\mathbb{I}$ značí jednotkovou matici a $\hat{\theta} = (\mathbb{X}^\mathsf{T}\mathbb{X})^{-1}\mathbb{X}^\mathsf{T}\mathbf{y}$. Označíme–li navíc libovolný řádek matice $\mathbb{X}$ jednoduše jako $X = \left(1, x, x^2, \ldots, x^p\right)$, potom distribuci (1.28) můžeme přepsat jednorozměrně

$$p(y|x) = \mathcal{N}\left(X \cdot \hat{\theta}, \sigma^2\right). \tag{1.29}$$

Tuto distribuci budeme později využívat v generativním modelování. Pokračujme tím, že určíme distribuci vektoru ϵ. To není nic těžkého, jelikož má každá složka stejné jednorozměrné Gaussovo rozdělení a také jsou všechny složky nezávislé. Z vlastností vícerozměrného Gaussova rozdělení [8] víme, že bude mít právě toto rozdělení, tedy

$$p(\epsilon) \propto \exp\left\{-\frac{1}{2}\epsilon^\mathsf{T}\epsilon\right\}, \tag{1.30}$$

kde pro jednoduchost $\sigma^2 = 1$. Dále z rovnice (1.8) získáme

$$\epsilon = \mathbf{y} - \mathbb{X} \cdot \theta \tag{1.31}$$

a transformujeme pomocí vztahu (1.15) z věty o transformaci náhodné veličiny, čímž získáme

$$p(\epsilon) = p(\mathbf{y}|\mathbb{X},\theta) \propto \exp\left\{-\frac{1}{2}(\mathbf{y} - \mathbb{X}\theta)^\mathsf{T}(\mathbf{y} - \mathbb{X}\theta)\right\}. \tag{1.32}$$

Poznámka. *Normalizační konstantu distribucí není nutno neustále psát, proto využíváme znak úměrnosti $\propto$, tj. rovnost až na jednoznačně určenou multiplikativní konstantu.*

Snažíme se získat distribuci $p(\theta|\mathbf{y},\mathbb{X})$, kterou získáme pomocí Bayesovy věty (1.13) následovně

$$p(\theta|\mathbf{y},\mathbb{X}) = \frac{p(\mathbf{y}|\mathbb{X},\theta)p(\theta|\mathbb{X})}{p(\mathbf{y}|\mathbb{X})} \propto p(\mathbf{y}|\mathbb{X},\theta)p(\theta|\mathbb{X}). \tag{1.33}$$

Zde můžeme na distribuci ve jmenovateli nahlížet jako na normalizační konstantu. K tomu abychom mohli pokračovat ve výpočtu $p(\theta|\mathbf{y},\mathbb{X})$, potřebujeme znát $p(\theta|\mathbb{X})$. Předpokládejme také, že je θ nezávislé na $\mathbb{X}$, budeme proto psát pouze $p(\theta)$. Pro $p(\theta)$ předpokládejme následující vztah

$$p(\theta) = \mathcal{N}\left(0, \alpha^{-1}\mathbb{I}\right) \propto \exp\left\{-\frac{1}{2}\theta^{\mathsf{T}}\theta\alpha\right\}. \tag{1.34}$$

Nyní můžeme pokračovat dosazením do (1.33) a obdržíme

$$\begin{aligned} p(\mathbf{y}|\mathbb{X},\theta)p(\theta|\mathbb{X}) &\propto \exp\left\{-\frac{1}{2}(\mathbf{y}-\mathbb{X}\theta)^{\mathsf{T}}(\mathbf{y}-\mathbb{X}\theta)\right\}\exp\left\{-\frac{1}{2}\theta^{\mathsf{T}}\theta\alpha\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left(\mathbf{y}^{\mathsf{T}}\mathbf{y} - \theta^{\mathsf{T}}\mathbb{X}^{\mathsf{T}}\mathbf{y} - \mathbf{y}^{\mathsf{T}}\mathbb{X}\theta + \theta^{\mathsf{T}}\mathbb{X}^{\mathsf{T}}\mathbb{X}\theta + \theta^{\mathsf{T}}\theta\alpha\right)\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left[\mathbf{y}^{\mathsf{T}}\mathbf{y} - \theta^{\mathsf{T}}\mathbb{X}^{\mathsf{T}}\mathbf{y} - \mathbf{y}^{\mathsf{T}}\mathbb{X}\theta + \theta^{\mathsf{T}}\left(\mathbb{X}^{\mathsf{T}}\mathbb{X} + \alpha\mathbb{I}\right)\theta\right]\right\}. \end{aligned} \tag{1.35}$$

Jedná se o součin dvou vícerozměrných Gaussovských rozdělení, proto můžeme předpokládat, že řešení bude ve tvaru kvadratické formy, která také odpovídá vícerozměrnému Gaussovu rozdělení. Tento tvar navíc obsahuje zbytek z po nejmenších čtvercích, ten ovšem také není nutné psát. Platí

$$p(\theta|\mathbf{y},\mathbb{X}) \propto \exp\left\{-\frac{1}{2}\left(\theta-\hat{\theta}\right)\Sigma^{-1}\left(\theta-\hat{\theta}\right) + z\right\} \propto \exp\left\{-\frac{1}{2}\left(\theta-\hat{\theta}\right)\Sigma^{-1}\left(\theta-\hat{\theta}\right)\right\}, \tag{1.36}$$

což upravujeme dále tak, abychom dokázali určit $\hat{\theta}$ a Σ. Roznásobením dostaneme

$$p(\theta|\mathbf{y},\mathbb{X}) \propto \exp\left\{-\frac{1}{2}\left(\theta^{\mathsf{T}}\Sigma^{-1}\theta - \hat{\theta}^{\mathsf{T}}\Sigma\theta - \theta^{\mathsf{T}}\Sigma^{-1}\hat{\theta} + \hat{\theta}^{\mathsf{T}}\Sigma^{-1}\hat{\theta}\right)\right\}, \tag{1.37}$$

z čehož už při porovnání výrazu $\theta^{\mathsf{T}}\left(\mathbb{X}^{\mathsf{T}}\mathbb{X} + \alpha\mathbb{I}\right)\theta$, nacházejícím se v konečném tvaru rovnice (1.35), s výrazem $\theta^{\mathsf{T}}\Sigma^{-1}\theta$ v předchozí rovnici (1.37), plyne předpis pro

$$\Sigma^{-1} = \mathbb{X}^{\mathsf{T}}\mathbb{X} + \alpha\mathbb{I}. \tag{1.38}$$

Přímo porovnávejme další dva výrazy z těchto rovnic

$$-\mathbf{y}^{\mathsf{T}}\mathbb{X}\theta = -\hat{\theta}^{\mathsf{T}}\Sigma^{-1}\theta. \tag{1.39}$$

Nyní z této rovnice jednoduchou úpravou a dosazením za Σ dostaneme předpis pro $\hat{\theta}$, a to

$$\hat{\theta} = \Sigma\mathbb{X}^{\mathsf{T}}\mathbf{y} = \left(\mathbb{X}^{\mathsf{T}}\mathbb{X} + \alpha\mathbb{I}\right)^{-1}\mathbb{X}^{\mathsf{T}}\mathbf{y}. \tag{1.40}$$

1.2.4 Divergence

Divergence [14] je funkce $D(.\|.) : S \times S \to \mathbb{R}$, kde S je prostor všech pravděpodobnostních distribucí a která navíc splňuje následující dvě podmínky

1. $D(q\|p) \geq 0$,
2. $D(q\|p) = 0 \quad \iff \quad p = q$.

Divergence do jisté míry popisuje vzdálenost nebo rozdíl mezi dvěma rozděleními. Jelikož divergence nemusí splňovat podmínku symetrie a trojúhelníkové nerovnosti, nejedná se tedy o metriku, nýbrž o semimetriku.

f-divergence

Nejdůležitější skupinou divergencí jsou takzvané f-divergence. Jsou definovány pomocí konvexní funkce $f(x)$, kde $x > 0$ a navíc pro ni platí podmínka $f(1) = 0$. Jsou tvaru

$$D_f(q\|p) = \int q(x) f\left(\frac{q(x)}{p(x)}\right) \mathrm{d}x. \tag{1.41}$$

Kullback-Leiblerova divergence

Pro nás bude nezbytně nutná tzv. Kullback-Leiblerova divergence, kde za funkci f bereme přirozený logaritmus. Ten je konvexní funkcí, pro kterou platí podmínka $\log 1 = 0$. Tvar KL-divergence je následující

$$D_{KL}(q\|p) = \int q(x) \log \frac{q(x)}{p(x)} \mathrm{d}x. \tag{1.42}$$

Divergence se často definují jednorozměrně, lze je ovšem alternativně definovat i ve více rozměrech.

1.2.5 ELBO

Předpokládejme, že máme $\mathbf{y}$ jako pozorování, dále $\mathbf{z}$ jsou latentní (skryté) proměnné, jinými slovy jsou to takové proměnné, které nejsou pozorovány přímo. Toto je zcela obecná definice a $\mathbf{z}$ tak může obsahovat i vektor parametrů. Posteriorní rozdělení latentní proměnné $\mathbf{z}$ můžeme napsat pomocí Bayesova pravidla (1.13) takto

$$p(\mathbf{z}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\mathbf{z})p(\mathbf{z})}{\int p(\mathbf{y}, \mathbf{z}) \mathrm{d}\mathbf{z}}. \tag{1.43}$$

Dále zadefinujeme nový objekt, věrohodnostní funkci jmenovatele

$$\log p(\mathbf{y}) = \log \int p(\mathbf{y}, \mathbf{z}) \mathrm{d}\mathbf{z}. \tag{1.44}$$

Jmenovatel v Bayesově pravidle se někdy také nazývá **evidence**. Abychom mohli pokračovat, využijeme pomocnou funkci $q(\mathbf{z}|\mathbf{w})$

$$\log p(\mathbf{y}) = \log \int p(\mathbf{y}, \mathbf{z}) \mathrm{d}\mathbf{z} = \log \int q(\mathbf{z}|\mathbf{w}) \frac{p(\mathbf{y}, \mathbf{z})}{q(\mathbf{z}|\mathbf{w})} \mathrm{d}\mathbf{z} = \log \mathbb{E}_q \left[\frac{p(\mathbf{y}, \mathbf{z})}{q(\mathbf{z}|\mathbf{w})} \right]. \tag{1.45}$$

Symbol $\mathbb{E}_q$ značí střední hodnotu přes $q(\mathbf{z}|\mathbf{w})$. Dále využijeme **Jensenovu nerovnost**, díky které získáme spodní hranici *(lower bound)*, odtud tedy **Evidence Lower Bound** [17], čili ELBO

$$\log \mathbb{E}_q \left[\frac{p(\mathbf{y}, \mathbf{z})}{q(\mathbf{z}|\mathbf{w})} \right] \geq \mathbb{E}_q \left[\log \frac{p(\mathbf{y}, \mathbf{z})}{q(\mathbf{z}|\mathbf{w})} \right] = L(\mathbf{w}). \tag{1.46}$$

ELBO je v tomto případě ztrátová funkce, proto ho značíme také $L(\mathbf{w})$. Dále rozepíšeme pomocí součinového pravidla (1.12), využijeme vlastností logaritmů a dle definice KL-divergence (1.42) přepíšeme do tvaru

$$L(\mathbf{w}) = \mathbb{E}_q \left[\log \frac{p(\mathbf{y}, \mathbf{z})}{q(\mathbf{z}|\mathbf{w})} \right] = \mathbb{E}_q \left[\log \frac{p(\mathbf{y}|\mathbf{z}) p(\mathbf{z})}{q(\mathbf{z}|\mathbf{w})} \right] = \mathbb{E}_q \left[\log p(\mathbf{y}|\mathbf{z}) \right] - \mathbb{E}_q \left[\log \frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{w})} \right] \tag{1.47}$$

$$= \mathbb{E}_q \left[\log p(\mathbf{y}|\mathbf{z}) \right] - D_{KL}(q(\mathbf{z}|\mathbf{w}) \| p(\mathbf{z})). \tag{1.48}$$

Budeme–li maximalizovat ELBO přes všechny variační parametry $\mathbf{w}$, získáme nejbližší možnou hodnotu k $\log p(\mathbf{y})$. Navíc je maximalizace ELBO ekvivalentní k minimalizaci KL-divergence mezi $q(\mathbf{z}|\mathbf{w})$ a $p(\mathbf{z}|\mathbf{w})$, jelikož platí

$$\begin{aligned} D_{KL}(q(\mathbf{z}|\mathbf{w}) \| p(\mathbf{z}|\mathbf{y})) &= \mathbb{E}_q \left[\log \frac{q(\mathbf{z}|\mathbf{w})}{p(\mathbf{z}|\mathbf{y})} \right] \\ &= \mathbb{E}_q \left[\log \frac{q(\mathbf{z}|\mathbf{w}) p(\mathbf{y})}{p(\mathbf{y}|\mathbf{z}) p(\mathbf{z})} \right] \\ &= -\mathbb{E}_q \left[\log p(\mathbf{y}|\mathbf{z}) \right] + \mathbb{E}_q \left[\log \frac{q(\mathbf{z}|\mathbf{w})}{p(\mathbf{z}|\mathbf{y})} \right] + \mathbb{E}_q \left[\log p(\mathbf{y}) \right] \\ &= -\mathbb{E}_q \left[\log p(\mathbf{y}|\mathbf{z}) \right] + D_{KL}(q(\mathbf{z}|\mathbf{w}) \| p(\mathbf{z})) + \log p(\mathbf{y}). \end{aligned} \tag{1.49}$$

Z toho jednoduchou úpravou dostaneme konečný vztah

$$D_{KL}(q(\mathbf{z}|\mathbf{w}) \| p(\mathbf{z}|\mathbf{y})) = -L(\mathbf{w}) + \log p(\mathbf{y}). \tag{1.50}$$

1.2.6 Příklad využití ELBO

Předvedeme příklad, jak ELBO využít v praxi. Uvažujme pouze sadu dvou pozorování y_1 a y_2 s Normálním rozdělením $\mathcal{N}_i(\theta, 1)$ pro $i \in \{1, 2\}$. Dále uvažujme jeden parametr $\theta|\alpha \sim \mathcal{N}(0, \alpha)$ a nechť α je tzv. **Jeffreyho prior** [11], tedy $\alpha \sim \text{invGamma}(0, 0+)$.

Snažíme se získat sdruženou distribuci parametrů θ a α, tedy $p(\theta, \alpha|y_1, y_2)$. Tuto distribuci můžeme přepsat pomocí definice podmíněné pravděpodobnosti (1.11) a řetězového pravidla (1.12) jako

$$p(\theta, \alpha|y_1, y_2) = \frac{p(\theta, \alpha, y_1, y_2)}{p(y_1, y_2)} = \frac{p(y_1|\theta) p(y_2|\theta) p(\theta) p(\alpha)}{p(y_1, y_2)}. \tag{1.51}$$

Abychom to uvedli do kontextu s definicí ELBO (1.47) - naše pozorování $\mathbf{y}$ je nyní vektor (y_1, y_2) a latentními proměnnými $\mathbf{z}$ rozumíme (α, θ). Dosazením předpokladů do čitatele dostaneme

$$p(y_1|\theta)p(y_2|\theta)p(\theta)p(\alpha) \propto \exp\left\{-\frac{1}{2}(y_1-\theta)^2\right\} \cdot \exp\left\{-\frac{1}{2}(y_2-\theta)^2\right\} \cdot \frac{1}{\sqrt{\alpha}} \exp\left\{-\frac{\theta^2}{2\alpha}\right\} \cdot \frac{1}{\alpha}. \quad (1.52)$$

Zdánlivě se nám může zdát určení jmenovatele jako jednoduché, protože distribuci $p(y_1, y_2)$ lze získat tzv. marginalizací, neboli vyintegrováním přes θ a α

$$\begin{aligned} p(y_1, y_2) &= \int p(\theta, \alpha, y_1, y_2)\mathrm{d}\theta\mathrm{d}\alpha \\ &= \int p(y_1|\theta)p(y_2|\theta)p(\theta)p(\alpha)\mathrm{d}\theta\mathrm{d}\alpha \\ &= \int \exp\left\{-\frac{1}{2}(y_1-\theta)^2\right\} \cdot \exp\left\{-\frac{1}{2}(y_2-\theta)^2\right\} \cdot \frac{1}{\sqrt{\alpha}} \exp\left\{-\frac{\theta^2}{2\alpha}\right\} \cdot \frac{1}{\alpha}\, \mathrm{d}\theta\mathrm{d}\alpha. \end{aligned} \quad (1.53)$$

Po bližším přezkoumání (1.53) zjistíme, že tuto distribuci nelze přes α vyintegrovat. Proto použijeme ELBO. Dle definice KL-divergence a za předpokladu nezávislosti $q(\theta, \alpha) = q(\theta)q(\alpha)$ můžeme psát

$$D_{KL}\left(q\left(\theta, \alpha|\mu, \sigma, \gamma, \delta\right) \| p(\theta, \alpha|y_1, y_2)\right) = \int q(\alpha)q(\theta) \log\left\{\frac{q(\alpha)q(\theta)}{p(\theta, \alpha|y_1, y_2)}\right\} \mathrm{d}\theta\mathrm{d}\alpha = \blacklozenge. \quad (1.54)$$

Nezapomínejme, že $q(\theta)$ a $q(\alpha)$ jsou distribuce, pro které zvolíme pochopitelný tvar o 4 neznámých parametrech $(\mu, \sigma, \gamma, \delta)$

$$\begin{aligned} q(\theta) &= \mathcal{N}(\mu, \sigma), \\ q(\alpha) &= \mathrm{invGamma}(\gamma, \delta). \end{aligned} \quad (1.55)$$

Zároveň to jsou naše variační parametry $\mathbf{w}$. Dle (1.2.4) víme, že platí $\int q(\alpha)q(\theta)\mathrm{d}\theta\mathrm{d}\alpha = 1$. Výraz budeme rozepisovat pomocí pravidel pro logaritmy a postupně upravovat. Navíc $p(y_1, y_2)$ v integrálu je konstanta, kterou můžeme pro jednoduchost zanedbat. Výsledek budeme na konci minimalizovat a konstanta polohu maxima nemění

$$\begin{aligned} \blacklozenge &= p(y_1, y_2) \cdot \int q(\alpha)q(\theta) \log \frac{q(\alpha)q(\theta)}{p(y_1|\theta)p(y_2|\theta)p(\theta)p(\alpha)} \mathrm{d}\theta\mathrm{d}\alpha \\ &\propto \int q(\theta)q(\alpha)\left(-\log p(y_1|\theta) - \log p(y_2|\theta) - \log p(\theta) - \log p(\alpha) + \log q(\theta) + \log q(\alpha)\right) \mathrm{d}\alpha\mathrm{d}\theta. \end{aligned} \quad (1.56)$$

Poslední dva výrazy jsou entropie pro Gaussovo rozdělení, resp. inverzní gamma rozdělení

$$\begin{aligned} \int q(\theta) \log q(\theta)\, \mathrm{d}\theta &\propto -\frac{1}{2}\log\sigma, \\ \int q(\alpha) \log q(\alpha)\, \mathrm{d}\alpha &= -\gamma - \log\left(\delta \cdot \Gamma(\gamma)\right) + (1+\gamma)\psi(\gamma). \end{aligned} \quad (1.57)$$

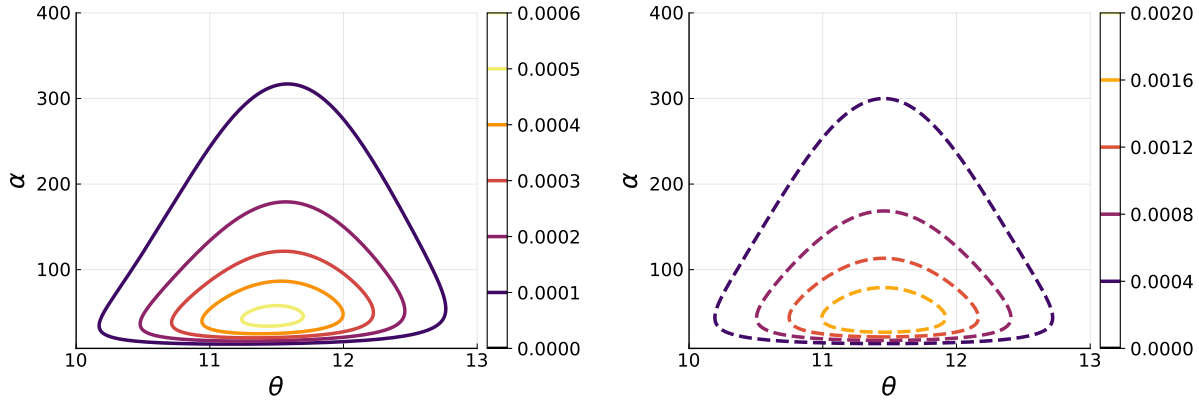
Vypočítejme zbývající výrazy, kde pro jednoduchost budeme pro střední hodnoty využívat značení pomocí špičatých závorek

$$\begin{aligned}
\int q(\theta)q(\alpha)\log p(y_1|\theta)\,\mathrm{d}\alpha\,\mathrm{d}\theta &= \left\langle -\frac{1}{2}(y_1-\theta)^2 \right\rangle = -\frac{1}{2}\left(y_1^2 - 2y_1\mu + \mu^2 + \sigma\right), \\
\int q(\theta)q(\alpha)\log p(y_2|\theta)\,\mathrm{d}\alpha\,\mathrm{d}\theta &= \left\langle -\frac{1}{2}(y_2-\theta)^2 \right\rangle = -\frac{1}{2}\left(y_2^2 - 2y_2\mu + \mu^2 + \sigma\right), \\
\int q(\theta)q(\alpha)\log p(\theta)\,\mathrm{d}\alpha\,\mathrm{d}\theta &= \left\langle -\frac{\theta^2}{2\alpha} - \frac{1}{2}\log\alpha \right\rangle = -\frac{1}{2}\left(\left(\mu^2+\sigma\right)\frac{\gamma}{\delta} + \log\delta - \psi(\gamma)\right), \\
\int q(\theta)q(\alpha)\log p(\alpha)\,\mathrm{d}\alpha\,\mathrm{d}\theta &= \langle -\log\alpha \rangle = \psi(\gamma) - \log\delta.
\end{aligned} \tag{1.58}$$

Nyní máme všechny potřebné výrazy pro výpočet odhadu parametrů $(\mu, \sigma, \gamma, \delta)$ distribuce $q(\theta, \alpha)$ numericky optimalizační metodou ADAM, tedy

$$\hat{\mu}, \hat{\sigma}, \hat{\gamma}, \hat{\delta} = \underset{\mu,\sigma,\gamma,\delta}{\arg\min}\, D_{KL}\left(q\left(\theta, \alpha|\mu, \sigma, \gamma, \delta\right) || p(\theta, \alpha|y_1, y_2)\right). \tag{1.59}$$

Porovnat obě distribuce můžeme na obrázku 1.1.



(a) Distribuce $p(\theta, \alpha|y_1, y_2)$

(b) Distribuce $q\left(\theta, \alpha|\mu, \sigma, \gamma, \delta\right)$

Obrázek 1.1: Contour plot distribuce $p(\theta, \alpha|y_1, y_2)$ (vlevo plnou čárou) a $q\left(\theta, \alpha|\mu, \sigma, \gamma, \delta\right)$ (vpravo čerchovaně), kde distribuce q je vyčíslena ve vypočtených odhadech $\hat{\mu}, \hat{\sigma}, \hat{\gamma}, \hat{\delta}$ a distribuce p je vyčíslena v bodech $y_1 = 11$ a $y_2 = 12$. Je patrné, že distribuce jsou téměř totožné.

1.3 Teorie grafů

Poslední teoretickou kapitolou bude vsuvka do teorie grafů [3]. Nepůjde nám však o grafy funkcí. Tato kapitola poslouží k výhodnému popsání složitějších datových struktur.

Definice 1.3.1 (Graf)**.** Grafem G se rozumí dvojice (V, H), kde V je množina vrcholů grafu G a H je množina hran tohoto grafu, přičemž jsou tyto množiny vzájemně disjunktní.

Toto je zcela obecná definice grafu. Takto definovaný graf je velmi silný nástroj ke zjednodušování složitých problémů. Je vhodný pro popis takových situací, jenž můžeme znázornit pomocí konečného množství bodů, čili vrcholů V a vztahů mezi nimi, které jsou znázorněny hranami H.

Definice 1.3.2 (Cesta v grafu)**.** Cestou v grafu rozumíme takovou posloupnost vrcholů a hran $(v_0, h_1, v_1, \ldots, h_t, v_t)$, kde vrcholy $v_0, \ldots, v_t$ jsou navzájem různé vrcholy grafu G a pro každé $i = 1, 2, \ldots, t$ je $e_i = \{v_{i-1}, v_i\} \in H$.

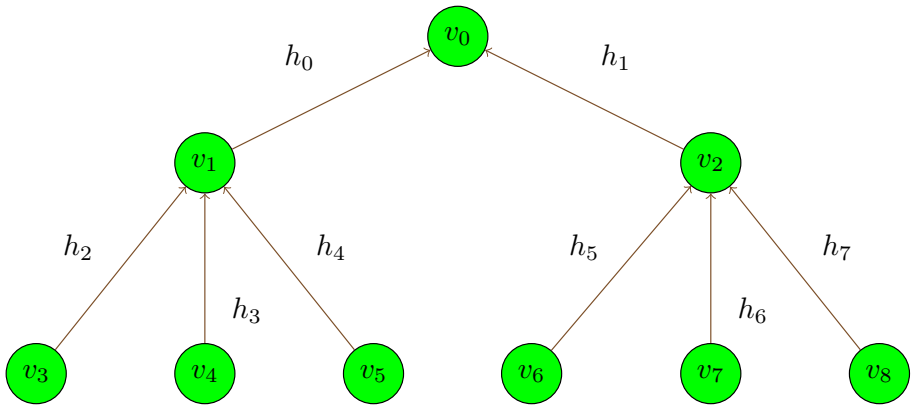
Definice 1.3.3 (Orientace v grafu)**.** Orientovaným grafem nazveme dvojici (V, H), kde H je podmnožina kartézského součinu $V \times V$. Prvky H pak nazýváme orientované hrany. Orientovaná hrana h je tvaru (x, y) a říkáme o ní, že vychází z x a končí v y.

Definice 1.3.4 (Souvislost grafu)**.** Řekneme, že graf G je souvislý, jestliže pro každé dva vrcholy v_0 a v_1 existuje v G cesta z v_0 do v_1.

Definice 1.3.5 (Cyklus v grafu)**.** Cyklem v grafu G rozumíme posloupnost vrcholů a hran $(v_0, h_1, v_1, \ldots, h_t, v_t = v_0)$, kde vrcholy $v_0, \ldots, v_{t-1}$ jsou navzájem různé vrcholy grafu G a pro každé $i = 1, 2, \ldots, t$ je $e_i = \{v_{i-1}, v_i\} \in H$.

Definice 1.3.6 (Strom)**.** Strom je souvislý graf neobsahující cyklus.

Takto definované grafy se často používají v diskrétní matematice. Primárním cílem je výhodně zadefinovat stromovou strukturu, budeme totiž pracovat s orientovanými stromy – ten můžeme vidět na obrázku 1.2. Jak na tyto definice napasovat generativní model bude hlavním cílem třetí kapitoly 3. Nejprve je nutno již zmíněný generativní model zadefinovat.



Obrázek 1.2: Příklad orientovaného stromu. Z definice stromu plyne, že mezi každými dvěma vrcholy existuje pouze jedna cesta a navíc platí, že počet vrcholů je o 1 větší, než počet hran.

Kapitola 2

Generativní modely

2.1 Generativní model

Ve strojovém učení se setkáváme s dvěma hlavními typy modelů a to jsou **generativní modely** a **diskriminativní modely** [12]. Každý z nich přistupuje k zadanému problému trochu jinak. Jak už napovídá název této práce, budeme se zde zabývat výhradně generativními modely.

Definice 2.1.1. (Generativní model) Mějme nějakou množinu datových záznamů $\mathbf{x} = \{x_1, \ldots, x_n\}$, představující nezávislé proměnné a nějakou množinu $\mathbf{y} = \{y_1, \ldots, y_n\}$, jakožto závislé proměnné. Generativní model je potom takový model, který se učí sdruženou distribuci $p(x, y)$.

Poznámka. *Diskriminativní modely se učí podmíněnou distribuci $p(y|x)$. Jsou využívány pro klasifikaci do tříd.*

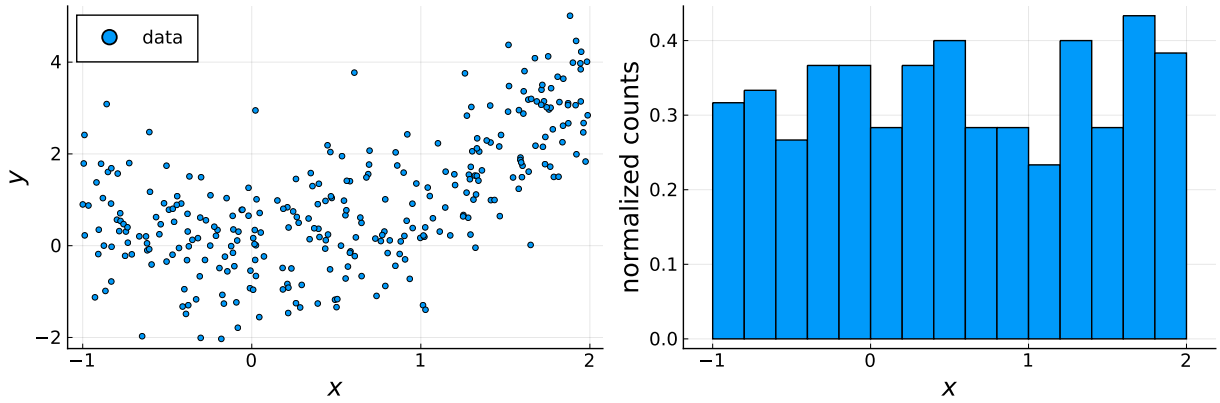
2.1.1 Příklad

Připomeňme nejprve, že platí $p(x, y) = p(y, x)$. V tomto okamžiku pro nás bude výhodnější hledat distribuci $p(y, x)$. Jeden ze způsobů jak odhadnout tuto distribuci, je využití součinového pravidla (1.12). Pomocí něj získáme tvar

$$p(y, x) = p(y|x) \cdot p(x). \tag{2.1}$$

Problém je tedy převeden na hledání distribucí $p(y|x)$ a $p(x)$. Pro ilustraci uvažujme množinu datových záznamů $\mathbf{x} = \{x_1, \ldots, x_n\}$ zobrazenou na obrázku 2.1.

1. Určíme distribuci $p(x)$. To není u tohoto příkladu nic problematického, můžeme ho určit například z histogramu x–ových souřadnic jednotlivých bodů, nebo použít maximálně věrohodný odhad [9] *(Maximum Likelihood Estimation, MLE)*. Data jsou na ose x, přesněji na intervalu (a, b), rozděleny rovnoměrně. To tedy indikuje hustotu rovnoměrného rozdělení
$$p(x) = \mathrm{U}(a, b). \tag{2.2}$$



(a) Data, pro která hledáme distribuci $p(x, y)$.

(b) Normalizovaný histogram vzorků z distribuce $p(x)$.

Obrázek 2.1: Data, pro která hledáme sdruženou distribuci $p(x, y)$ s histogramem distribuce $p(x)$. Z obrázku (*a*) je patrné, že rozdělení $p(x)$ je rovnoměrné, datové záznamy se na ose x totiž nikde neshlukují.

2. Nyní přejdeme k hledání distribuce $p(y|x)$. Tu můžeme určit pomocí metody nejmenších čtverců (1.29), protože víme že pro takovou distribuci platí

$$p(y|x) = \mathcal{N}\left(X \cdot \hat{\theta}, \sigma^2\right), \tag{2.3}$$

kde $X = \left(1, x, x^2\right)$, jelikož předpokládáme, že se jedná o kvadratickou závislost.

Nyní máme obě složky, můžeme tak zapsat konečný tvar sdružené distribuce

$$p(y, x) = \mathcal{N}\left(X \cdot \hat{\theta}, \sigma^2\right) \cdot \mathrm{U}(a, b). \tag{2.4}$$

V tomto okamžiku jsme již schopni generovat nová data. Na obrázku 2.2 vidíme contour plot této distribuce.

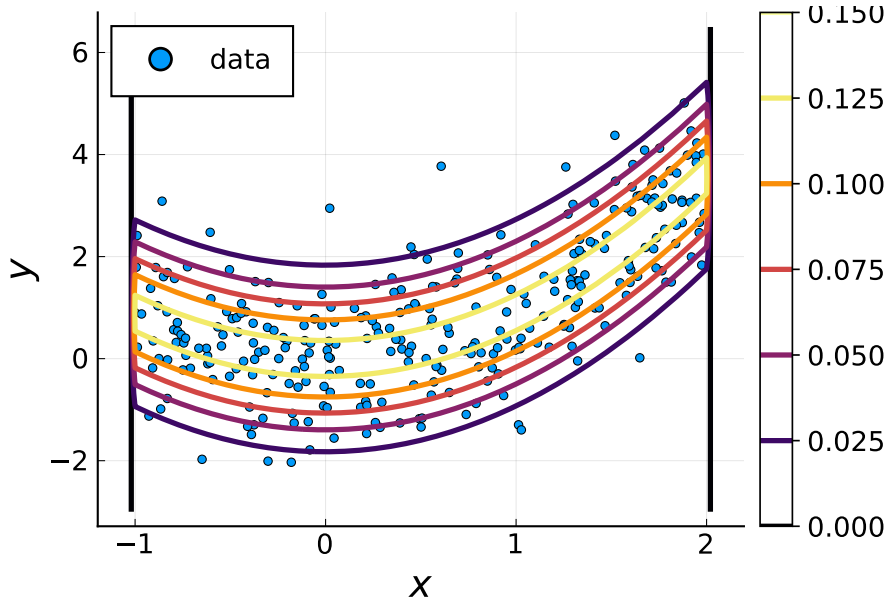
2.2 Neuronová síť

Nejjednodušší model je takový, který obsahuje pouze lineární kombinaci vstupních proměnných $(x_1, \ldots, x_n)$, tedy lineární model

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = w_0 + \sum_{i=1}^{n-1} w_i x_i. \tag{2.5}$$

Nyní se pokusíme tento model rozšířit tím, že do něj vneseme nelineární funkce vstupních proměnných a celé to obalíme do nelineární aktivační funkce f, čímž získáme novou funkci

$$y(\mathbf{x}, \mathbf{w}) = f\left(w_0 + \sum_{j=1}^{m} w_j \phi_j(\mathbf{x})\right). \tag{2.6}$$



Obrázek 2.2: Data z obrázku 2.1, tentokrát vyobrazená s contour plotem distribuce $p(y, x)$, kde $\sigma^2 = 1$, $a = -1$, $b = 2$ a y závisí na x kvadraticky. Distribuce je na krajích useknutá kvůli uniformnímu rozdělení – kdyby byla distribuce $p(x)$ Gaussovská, byla by výsledná distribuce na okrajích zakulacená.

Funkce $\phi_j(\mathbf{x})$ nazýváme **bázové funkce**. Parametr w_0 nám dovoluje nastavit offset, neboli tzv. práh *(bias)* v daných datech. Nyní představíme koncept neuronové sítě, který může být popsán sérií funkčních transformací. Nejprve zkonstruujeme m lineárních kombinací vstupních proměnných $(x_1, \ldots, x_n)$ ve tvaru

$$a_j = \sum_{i=1}^{n} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \tag{2.7}$$

kde j nabývá hodnot z $\{1, \ldots, m\}$ a horní index $^{(1)}$ značí, že příslušné parametry jsou v první vrstvě. Parametry $w_{ji}^{(1)}$ budeme nazývat váhy *(weights)* a $w_{j0}^{(1)}$ jsou složky již zmiňovaného prahu. Objekty a_j budeme nazývat aktivace *(activation)*, každou aktivaci transformujeme pomocí diferencovatelné, nelineární aktivační funkce h a dostaneme

$$z_j = h\left(a_j\right). \tag{2.8}$$

Z předchozího textu jasně plyne, že tento objekt odpovídá objektu popsánému funkcí (2.6). V kontextu neuronových sítí budeme tyto objekty nazývat skryté jednotky *(hidden units)*, proto tedy to intuitivní značení. Nyní budeme pokračovat ve stejném postupu, vezmeme hodnoty z_j, opět je lineárně zkombinujeme a získáme

$$a_k = \sum_{j=1}^{m} w_{kj}^{(2)} z_j + w_{k0}^{(2)}, \tag{2.9}$$

kde k nabývá hodnot z $\{1, \ldots, l\}$, zároveň l značí celkový počet výstupů a podobně jako předtím, horní index $^{(2)}$ značí, že příslušné parametry jsou ve druhé vrstvě. Aktivace rovněž jako

v předchozím kroku obalíme do další aktivační funkce f a získáme finální výstup

$$y_k = f\left(a_k\right). \tag{2.10}$$

Aktivační funkci jsme označili f místo h, poněvadž už dané jednotky nejsou skryté, ale jedná se v našem případě o výstup.
Teď se ovšem pokusme pojem aktivační funkce trochu více specifikovat. Volba aktivační funkce je různá případ od případu a záleží čistě na datech, na předpokládaném tvaru distribuce výstupních dat, atd. Existuje jich tedy mnoho, pro ilustraci předvedeme několik příkladů

$$\begin{aligned}
\text{Identita:} \qquad & f(x) = x, \\
\text{Sigmoidální:} \qquad & f(x) = \sigma\left(x\right) = \frac{1}{1+\exp\left(-x\right)}, \\
\text{Swish:} \qquad & f(x) = x \cdot \sigma\left(x\right) \\
\text{ReLU } \textit{(Rectified Linear Unit)}: \qquad & f(x) = \begin{cases} x & \text{pro } x \geq 0 \\ 0 & \text{pro } x < 0 \end{cases}, \\
\text{SELU } \textit{(Scaled Exponential Linear Unit)}: \qquad & f(x) = \lambda \cdot \begin{cases} x & \text{pro } x \geq 0 \\ \alpha\left(\mathrm{e}^x - 1\right) & \text{pro } x < 0 \end{cases}.
\end{aligned} \tag{2.11}$$

Pokud tedy spojíme všechny tyto kroky a zvolíme–li například sigmoidální tvar aktivační funkce σ, dostaneme tvar dvouvrstvé neuronové sítě [1]

$$y_k\left(\mathbf{x}, \mathbf{w}\right) = \sigma\left(\sum_{j=1}^{m} w_{kj}^{(2)} \cdot h\left(\sum_{i=1}^{n} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right). \tag{2.12}$$

Všechny váhy a složky prahu byly umístěny do vektoru vah $\mathbf{w}$. Neuronová síť je jednoduše řečeno pouze nelineární funkce z množiny vstupních proměnných $\{x_i\}_{i=1}^{n}$ do množiny $\{y_k\}_{k=1}^{l}$, určená vektorem $\mathbf{w}$. Na množství vrstev v neuronové síti se meze nekladou, alternativně lze sestrojovat další a další vrstvy.

2.3 Variační autoencoder

Jedna z mnoha metod, jak využít neuronové sítě, je metoda variačního autoencoderu (dále jen VAE) [5], [7]. Cílem je najít distribuce $p(\mathbf{x})$ vzorků $\{x_i\}_{i=1}^{n}$. Předpokládáme následující vztahy

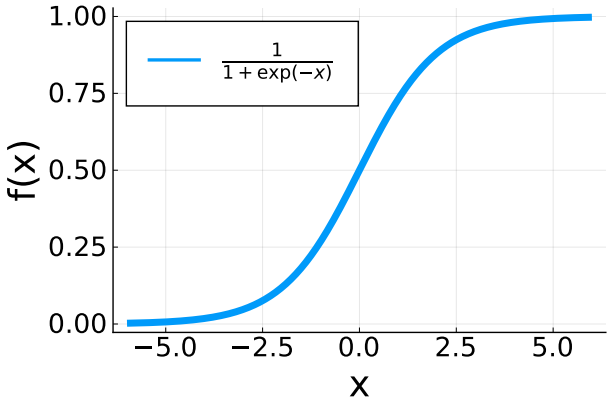
$$\mathbf{x} = f_\theta(\mathbf{z}) + \epsilon, \tag{2.13}$$

kde $\epsilon \sim \mathcal{N}\left(0, \sigma^2 \cdot \mathbb{I}\right)$ a $f_\theta(\mathbf{z})$ je neuronová síť. Využijeme následující formu aproximace

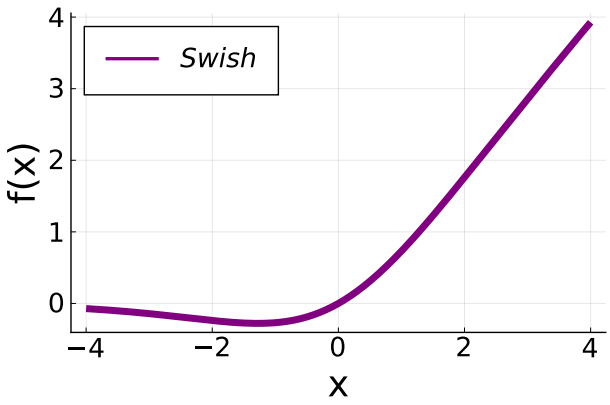
$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) \mathrm{d}\mathbf{z}. \tag{2.14}$$

Podle vztahu pro $\mathbf{x}$ určíme distribuce $p\left(\mathbf{x}|\mathbf{z}\right)$ a $p\left(\mathbf{z}\right)$ zvolíme jednoduše

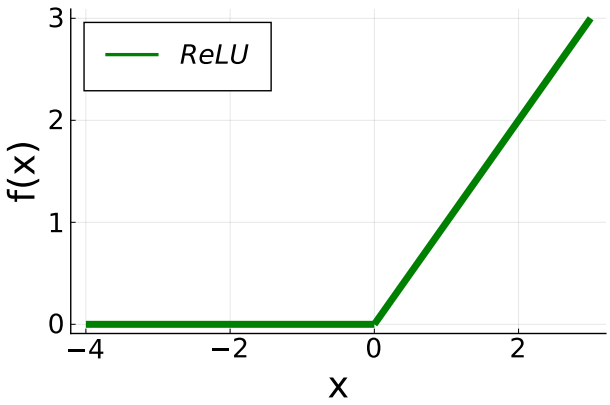
$$\begin{aligned}
p(\mathbf{x}|\mathbf{z}) &= \mathcal{N}\left(f_\theta(\mathbf{z}), \sigma^2 \cdot \mathbb{I}\right), \\
p(\mathbf{z}) &= \mathcal{N}\left(0, \mathbb{I}\right).
\end{aligned} \tag{2.15}$$



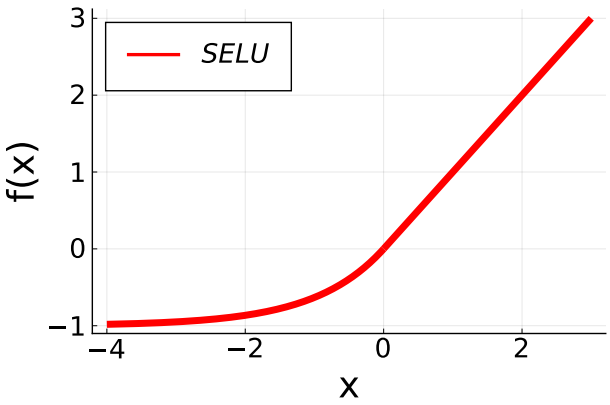
(a) Sigmoidální aktivační funkce



(b) Aktivační funkce Swish



(c) Aktivační funkce ReLU



(d) Aktivační funkce SELU

Obrázek 2.3: Aktivační funkce neuronové sítě. Nekladou se na ně žádné podmínky, můžou to být funkce omezené či neomezené, monotónní či periodické, eventuálně nemusí být ani spojité (jednotkový skok).

2.3.1 Naivní přístup

K nalezení $p(\mathbf{x})$ je třeba najít parametry θ transformace $f_\theta(\mathbf{z})$, proto zkusme sestavit věrohodnostní funkci $\log p(\mathbf{x}) = \log \prod_{i=1}^{n} p(x_i)$ a minimalizovat

$$\begin{aligned}
\hat{\theta} &= \arg\min_\theta \sum_{i=1}^{n} \log p(x_i) \\
&= \arg\min_\theta \sum_{i=1}^{n} \log \int \mathcal{N}\left(f_\theta(z_j), \sigma^2\right) \cdot \mathcal{N}(0,1)\, \mathrm{d}z_j \\
&= \arg\min_\theta \sum_{i=1}^{n} \log \sum_{j=1}^{n} \exp\left\{-\frac{1}{2\sigma^2}\left(x_i - f_\theta(z_j)\right)^2\right\} \cdot \exp\left\{-\frac{z_j^2}{2}\right\}.
\end{aligned} \tag{2.16}$$

Integrace přes $\mathbf{z}$ je nahrazena vzorkováním. Tento postup ovšem při minimalizaci nemusí konvergovat ke správným výsledkům.

2.3.2 Variační Bayesova metoda

Lepší metodou se ukazuje vzorkovat z podmíněné distribuce $q(\mathbf{z}|\mathbf{x})$ a využít ELBO

$$\begin{aligned} D_{KL}\left(q\left(\mathbf{z}|\mathbf{x}\right)||p(\mathbf{z}|\mathbf{x})\right) &= \mathbb{E}_q\left[\log q(\mathbf{z}|\mathbf{x}) - \log p\left(\mathbf{z}|\mathbf{x}\right)\right] \\ &= \mathbb{E}_q\left[\log q(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{x}|\mathbf{z}) - \log p(\mathbf{z}) + \log p(\mathbf{x})\right]. \end{aligned} \tag{2.17}$$

Tuto rovnici můžeme přepsat pomocí KL–divergence

$$\log p(\mathbf{x}) - D_{KL}\left(q\left(\mathbf{z}|\mathbf{x}\right)||p(\mathbf{z}|\mathbf{x})\right) = \mathbb{E}_q\left[\log p(\mathbf{x}|\mathbf{z})\right] - D_{KL}\left(q\left(\mathbf{z}|\mathbf{x}\right)||p(\mathbf{z})\right), \tag{2.18}$$

kde pravá strana této rovnice je lower bound objektu $\log p(\mathbf{x})$. Jestliže vybereme parametrickou formu distribuce

$$q\left(\mathbf{z}|\mathbf{x}\right) = \mathcal{N}\left(\mu_\phi(\mathbf{x}), \operatorname{diag}\left(\sigma_\phi^2(\mathbf{x})\right)\right), \tag{2.19}$$

můžeme parametry θ a ϕ minimalizovat zároveň a to následovně

$$\begin{aligned} \hat{\theta}, \hat{\phi} &= \arg\min_{\theta,\phi} \sum_{i=1}^{n} \log p\left(x_i\right) \\ &= \arg\min_{\theta,\phi} \left\{\mathbb{E}_q\left[\log p(\mathbf{x}|\mathbf{z})\right] - D_{KL}\left(q\left(\mathbf{z}|\mathbf{x}\right)||p(\mathbf{z})\right)\right\}. \end{aligned} \tag{2.20}$$

V metodě variačního autoencoderu jsou nezbytné následující dva fakty.

1. Trik v **reparametrizaci**

   $$\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \epsilon, \tag{2.21}$$

   kde $\odot$ značí Hadamardův součin, čili součin po složkách. To můžeme zapsat jednodušeji takto

   $$z_i = \mu_\phi(x_i) + \sigma_\phi(x_i) \cdot \epsilon_i. \tag{2.22}$$

   Nejedná se v podstatě o nic jiného, než o transformaci náhodné veličiny.

2. KL–divergence dvou Gaussovských distribucí má analytické řešení a nabude tvaru

   $$\begin{aligned} D_{KL}\left(q\left(\mathbf{z}|\mathbf{x}\right)||p(\mathbf{z})\right) &= \frac{1}{2}\left[\operatorname{tr}\left(\operatorname{diag}\left(\sigma_\phi^2(\mathbf{x})\right)\right) - \mu_\phi^\intercal(\mathbf{x})\mu_\phi(\mathbf{x}) - k - \log\det\operatorname{diag}\left(\sigma_\phi^2(\mathbf{x})\right)\right] \\ &= \frac{1}{2}\left[\sum_{l=1}^{k}(\sigma_\phi^2(\mathbf{x})) - \mu_\phi^\intercal(\mathbf{x})\mu_\phi(\mathbf{x}) - k - \sum_{l=1}^{k}\log\sigma_\phi^2(\mathbf{x})\right], \end{aligned} \tag{2.23}$$

   kde k značí dimenzi Gaussova rozdělení.

Kdybychom totiž nevybrali aproximační distribuce Gaussovské, nemohli bychom tímto způsobem $\hat{\theta}, \hat{\phi}$ určit. Díky těmto dvěma faktům tak získáme konečný tvar odhadu parametrů

$$\hat{\theta}, \hat{\phi} = \arg\min_{\theta,\phi} \sum_{i=1}^{n} \sum_{j=1}^{p} \left[x_i - f_\theta \left(\mu_\phi(x_i) + \sigma_\phi(x_i) \cdot \epsilon_{i,j} \right) \right]^2 - \frac{1}{2} \left[\sum_{l=1}^{k} (\sigma_\phi^2(x_i)) - \mu_\phi^\intercal(x_i) \mu_\phi(x_i) - k - \sum_{l=1}^{k} \log \sigma_\phi^2(x_i) \right]. \quad (2.24)$$

Poznámka. *Ve variační Bayesově metodě je nezbytné správně zvolit rozdělení latentní proměnné, v našem případě to jsou rozdělení $p(\boldsymbol{z})$ a $q(\boldsymbol{z}|\boldsymbol{x})$. V rovnici* (2.15) *jsme stanovili předpoklad pro*

$$p(\boldsymbol{z}) = \mathcal{N}(0, \mathbb{I}),$$

což byla čistě naše volba. Z tohoto důvodu jsme také museli zvolit rozdělení $q(\boldsymbol{z}|\boldsymbol{x})$ jako Gaussovské, kvůli analytičnosti KL–diveregence. Není to ovšem jediné řešení, jak rozdělení latentní proměnné zvolit. Fungovala by jakákoliv dvě rozdělení, jejichž KL-divergence má analytické řešení. Chceme–li však analytickou KL–divergenci, tak by měla být obě rozdělení stejná, vždy však musíme zvážit, je–li $q(\boldsymbol{z}|\boldsymbol{x})$ vhodnou aproximací aposteriorní distribuce. Analytické řešení KL–divergence dostaneme např. pro dvojici: Gamma rozdělení, Poissonova rozdělení nebo Bernoulliho rozdělení.

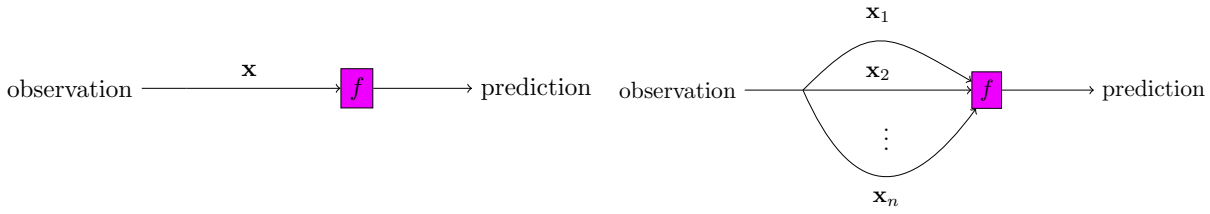
Kapitola 3

Stromové struktury

Stromovou strukturou dat rozumíme množinu datových záznamů popsaných pomocí množiny vrcholů a hran. Vrcholy dané stromové struktury představují jednotlivé body x a y. V podstatě si to můžeme představit opravdu jako strom – má jeden kořen, v první úrovni se dělí na k_1 větví, každá další větev se v druhé úrovni dělí na k_{2_i} a tak dále. My se v této práci budeme zabývat pouze kořenem a první úrovní větví.

3.1 Multi–instanční učení

Multi–instanční učení *(Multiple Instance Learning, MIL)* [2] se od klasického strojového učení liší tím, že každý vzorek je popsán pomocí množiny vektorů hodnot $b = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, zatímco u klasického strojového učení je daný vzorek popsán jedním vektorem hodnot $\mathbf{x}$. Tuto množinu vektorů $\mathbf{x} \in \mathcal{X}$ nazýváme pluk *(bag)* a jednotlivé vektory nazýváme instance *(instances)*, kde $\mathcal{X}$ je prostor všech instancí. Velikost této množiny, značeno $|b|$, může nabývat jakéhokoliv přirozeného čísla včetně 0. Pluky náleží prostoru pluků *(bag space)* $\mathcal{B} = \mathcal{P}_F(\mathcal{X})$, kde symbol $\mathcal{P}_F(\mathcal{X})$ značí všechny konečné podmnožiny $\mathcal{X}$.



(a) Standardní strojové učení

(b) Multi-instanční učení

Obrázek 3.1: Rozdíl mezi standardním strojovým učením a multi–instanční učením [2]. Standardní strojové učení je tedy speciální případ multi–instačního učení s velikostí pluku $|b| = 1$.

3.1.1 Vnořený prostor a agregační funkce

Metody využívající vnořeného prostoru *(Embedded Space, ES)* [2] definují vektorový prostor a vymezují mapovaní *(mapping)* z každého pluku $b \in \mathcal{B}$ do tohoto prostoru. Je–li tento prostor $\mathbb{R}^m$ a označíme–li jednu mapovací funkci $\phi_i : \mathcal{B} \mapsto \mathbb{R}$, $i \in \{1, \ldots, m\}$, celkové **vnoření** je pak funkce $\boldsymbol{\phi} : \mathcal{B} \mapsto \mathbb{R}^m$, kterou zapisujeme následovně

$$\boldsymbol{\phi}(b) = (\phi_1(b), \phi_2(b), \ldots, \phi_m(b)). \tag{3.1}$$

Mapovací funkce ϕ_i má za cíl získat a vhodně agregovat informace ze všech instancí, proto je definujeme způsobem

$$\phi_i(b) = g(\{k(\mathbf{x})\}_{\mathbf{x} \in b}), \tag{3.2}$$

kde $k : \mathcal{X} \mapsto \mathbb{R}^m$ značí transformaci instance a $g : \mathcal{P}_F(\mathbb{R}^m) \mapsto \mathbb{R}^d$ je **agregační funkce**, přičemž d bývá většinou vyšší dimenze než m. To zapříčiní, že na takto definovaný objekt již můžeme používat standardní algoritmy strojového učení.

Poznámka. *Nejčastěji používané agregační funkce jsou minimum, maximum, počet, či průměr.*

3.1.2 Jednoduchý příklad

Mějme množinu pozorování $\mathbf{y} = \{y_1, \ldots, y_n\}$. Předpokládejme takový model, který ke každému $y_i \in \mathbf{y}$, přiřazuje vektor datových záznamů $\mathbf{x}_i$, $i \in \{1, \ldots, n\}$, přičemž každý $\mathbf{x}_i$ má různý počet prvků $x_j^{(i)}$, $j \in \left\{1, \ldots, N_x^{(i)}\right\}$. Máme tedy $b = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, kde každý vektor $\mathbf{x}_i$ může mít jiný počet prvků $N_x^{(i)}$. Celkový počet datových záznamů v každé instanci množiny b je m, platí tedy $\sum_{i=1}^{n} \sum_{j=1}^{N_x^{(i)}} x_j^{(i)} = m$. Přiřazení probíhá následujícím způsobem

$$\begin{aligned} \mathbf{x}_1 &\longmapsto y_1, \\ \mathbf{x}_2 &\longmapsto y_2, \\ &\vdots \\ \mathbf{x}_n &\longmapsto y_n. \end{aligned} \tag{3.3}$$

Jednoduše řečeno je to přiřazení popořadě. Abychom to uvedli do kontextu stromových struktur, znamená to, že y_i jsou uzly jednoho typu, $x_j^{(i)}$ jsou uzly druhého typu, mezi nimiž existují hrany, čili funkce, které tyto hrany spojují. Počet prvků v jedné instanci $N_x^{(i)}$ nechť je generován například pomocí Poissonova rozdělení, tedy

$$p\left(N_x^{(i)}\right) = \mathrm{Po}(\lambda) \quad \forall i \in \{1, \ldots, n\}. \tag{3.4}$$

Potom všechny prvky každé instance $\mathbf{x}_i$ nechť jsou například generovány pomocí uniformního rozdělení

$$p\left(x_j^{(i)}\right) = \mathrm{U}(a, b) \quad \forall i \in \{1, \ldots, n\} \quad \& \quad \forall j \in \left\{1, \ldots, N_x^{(i)}\right\}. \tag{3.5}$$

Potom $\mathbf{y}$ nechť závisí na projekci $\mathbf{x}_i$ do vnořeného prostoru, tj. na výsledku agregačních funkcí aplikovaných na vstupní stromovou strukturu. Pokusme se nalézt sdruženou distribuci $p(y, \bar{x}_i)$,

kde $\bar{x}_i$ značí agregační funkci aritmetický průměr prvků v i–té instanci, čili

$$\bar{x}_i = \frac{1}{N_x^{(i)}} \sum_{l=1}^{N_x^{(i)}} x_l^{(i)}. \tag{3.6}$$

Použijeme opět součinové pravidlo

$$p(y, \bar{x}_i) = p(y|\bar{x}_i) \cdot p(\bar{x}_i) \tag{3.7}$$

a pokusíme se nalézt tyto dvě distribuce. Tento postup je nám známý už z kapitoly (2).

1. Pro určení podmíněné distribuce $p(y|\bar{x}_i)$ použijeme opět metodu nejmenších čtverců a dostaneme
$$p(y|\bar{x}_i) = \mathcal{N}\left(X \cdot \hat{\theta}, \sigma^2\right). \tag{3.8}$$
Ovšem zde jsou prvky vektoru X právě aritmetické průměry, tedy
$$X = \left(1, \bar{x}, \bar{x}^2, \ldots, \bar{x}^p\right). \tag{3.9}$$

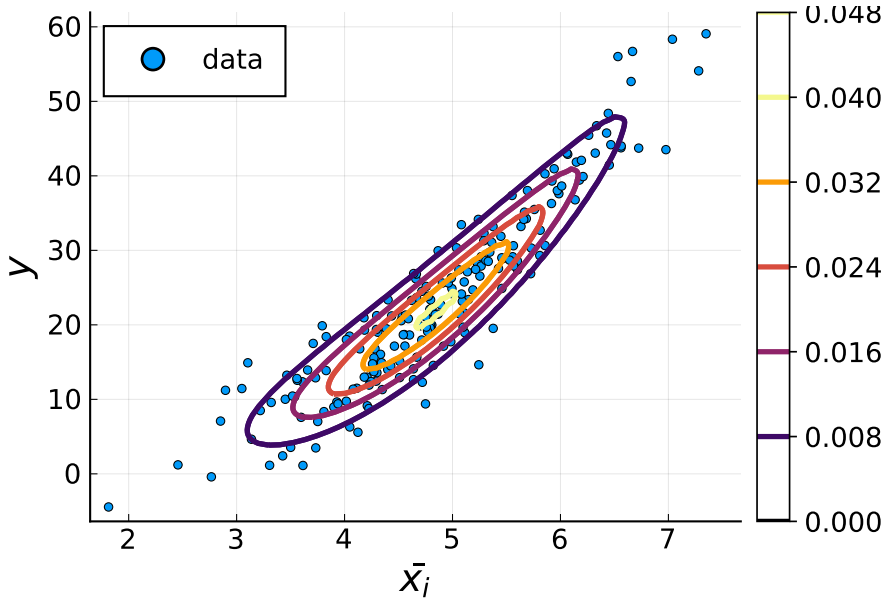
2. Distribuci $p(\bar{x}_i)$ lze určit obdobně pomocí histogramu. Navíc víme–li, že se jedná o výběrové průměry, je z centrální limitní věty [8] jasné, že se bude jednat o Gaussovo rozdělení. Střední hodnotu můžeme odhadnout výběrovým průměrem z $\bar{x}_i$, tedy
$$\bar{\bar{x}} = \frac{1}{n} \sum_{i=1}^{n} \bar{x}_i \tag{3.10}$$
a rozptyl odhadneme pomocí výběrového rozptylu
$$\text{Var}\ (\bar{x}_i) = \frac{1}{n} \sum_{i=1}^{n} (\bar{x}_i - \bar{\bar{x}})^2. \tag{3.11}$$
Oba objekty jsou maximálně věrohodnými odhady Gaussova rozdělení. Získáme tak tvar
$$p(\bar{x}_i) = \mathcal{N}\left(\bar{\bar{x}}_i, \text{Var}\ (\bar{x}_i)\right). \tag{3.12}$$

Tímto máme spočtené obě složky a finální podobu

$$p(y, \bar{x}_i) = \mathcal{N}\left(X \cdot \hat{\theta}, \sigma^2\right) \cdot \mathcal{N}\left(\bar{\bar{x}}_i, \text{Var}\ (\bar{x}_i)\right). \tag{3.13}$$

Pro vizualizaci této distribuce využijeme opět contour plot, viz obrázek 3.2.

Poznámka. *Jedná se o součin dvou Gaussovských distribucí. Očekáváme, že contour plot výsledné distribuce bude kulatý a nebude mít žádné ostré hrany. Najít distribuci $p(y|x)$ není v tomto případě snadný úkol, proto jsme se omezili pouze na hledání distribuce z průměrů jednotlivých instancí pluku b.*



Obrázek 3.2: Countour plot distribuce $p(y, \bar{x}_i)$, kde $m = 200, \lambda = 10, a = 0, b = 10$ a y závisí na $\bar{x}_i$ kvadraticky.

Řešení příkladu 3.1.2 pomocí VAE

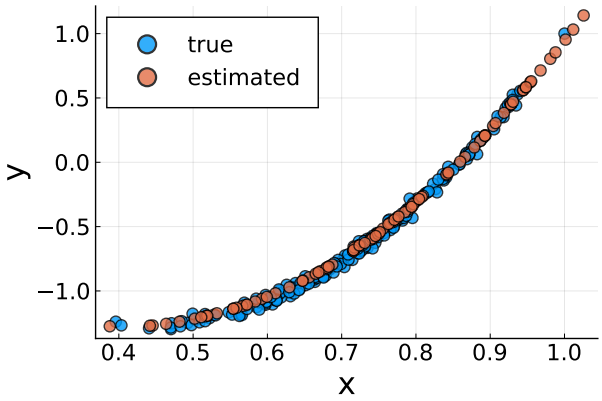
Podívejme se nyní na to, jak najít řešení jednoduchých stromových struktur pomocí VAE. Příklad je totožný s předchozím, máme pouze jinou kvadratickou závislost a využíváme neuronovou síť. Hledáme parametry θ transformační funkce f_θ. Stejně tak hledáme její inverzní funkci. My hledáme takovou funkci, která nám dokázala transformovat vzorky generované z $\mathcal{N}(0, 1)$ do dvou rozměrů. Funkčnost ověříme tak, že vykreslíme skutečné vzorky $\{\mathbf{x}, \mathbf{y}\}$ společně se vzorky $f_\theta(\mathbf{z})$. Její inverzní funkce tudíž dokáže transformovat vzorky $\{\mathbf{x}, \mathbf{y}\}$ zpátky na vzorky z $\mathcal{N}(0, 1)$ – to ověříme pomocí histogramu těchto vzorků a pro porovnání vykreslíme skutečné rozdělení $\mathcal{N}(0, 1)$, ke které by se měl histogram blížit. Výsledky jsou vykresleny na obrázku 3.3. Pro natrénování tohoto modelu bylo třeba zvolit aktivační funkci SELU.

3.1.3 Příklad se směsovým modelem

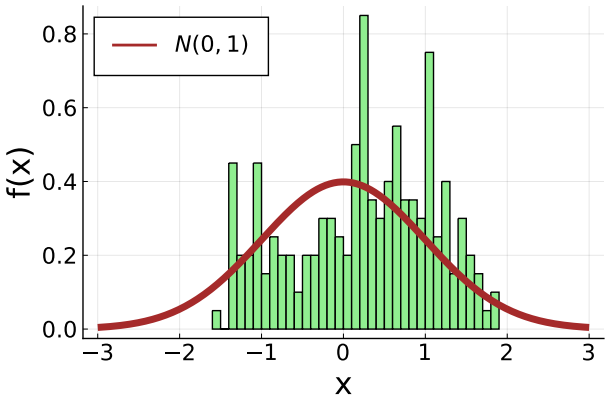
Dalším příkladem stromové struktury může být následující model, inspirovaný finanční aplikací. Veličina x_j udává hodnotu transakce na bankovních účtech klientů a instance $\mathbf{x}_i$ označuje všechny transakce klienta za sledované období. Budou to dvě Gaussovské směsi *(Gaussian Mixture, GM)*

$$\begin{aligned} p(x_i|y = 1) &= w_1 \cdot \mathcal{N}\left(\mu_1, \sigma_1^2\right) + (1 - w_1) \cdot \mathcal{N}\left(\mu_2, \sigma_2^2\right), \\ p(x_j|y = 0) &= w_2 \cdot \mathcal{N}\left(\mu_3, \sigma_3^2\right) + (1 - w_2) \cdot \mathcal{N}\left(\mu_4, \sigma_4^2\right), \end{aligned} \tag{3.14}$$

přičemž z první máme a a z druhé b vzorků. Podmínka y je teď diskrétní náhodná veličina nabývajících pouze dvou hodnot z množiny $\{0, 1\}$. Zároveň je to veličina udávající, zda je klient schopen splácet půjčku, kde $y = 0$ znamená *není schopen splácet* a $y = 1$ znamená



(a) Skutečné vzorky $\{\mathbf{x}, \mathbf{y}\}$ (modře) a jejich odhad pomocí VAE (červeně).



(b) Histogram vzorků $\mathbf{z}$, určených pomocí inverzní transformační funkce.

Obrázek 3.3: Ukázka funkčnosti VAE na jednoduchých stromových strukturách příkladu 3.1.3.

je schopen splácet. Nakonec $w \in [0, 1]$ značí váhu. Dále budeme uvažovat počty transakcí N_x daného klienta a distribuce

$$\begin{aligned} p(N_x^{(1)}|y = 1) &= \mathrm{Po}\,(\lambda_1)\,, \\ p(N_x^{(0)}|y = 0) &= \mathrm{Po}\,(\lambda_2)\,, \end{aligned} \tag{3.15}$$

což tedy udává takové rozdělení počtů transakcí klienta, jestli je schopen nebo není schopen splácet půjčku.
Jak takové distribuce odhadnout? Jelikož známe jejich tvar, stačí odhadnout pouze parametry a to například pomocí MLE. Sestavíme věrohodnostní funkce

$$\begin{aligned} \ell\left(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, w_1\right) &= \log\left(\prod_{i=1}^{a} p(x_i|y = 1)\right), \\ \ell\left(\mu_3, \mu_4, \sigma_3^2, \sigma_4^2, w_2\right) &= \log\left(\prod_{j=1}^{b} p(x_j|y = 0)\right). \end{aligned} \tag{3.16}$$

Obdobně bychom sestavili věrohodnostní funkce i pro Poissonovo rozdělení. Věrohodnostní funkce opět numericky maximalizujeme pomocí optimalizační metody ADAM a získáme

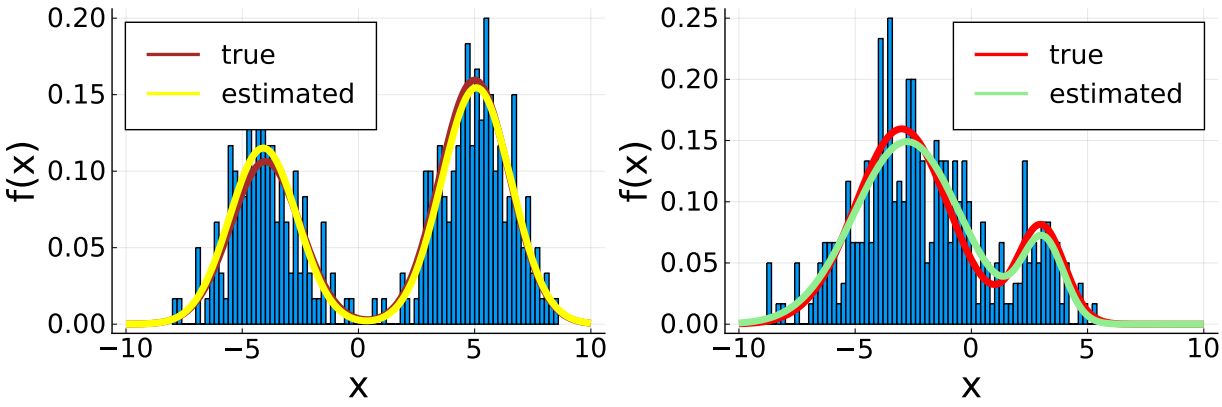
$$\begin{aligned} \hat{\mu_1}, \hat{\mu_2}, \hat{\sigma_1}^2, \hat{\sigma_2}^2, \hat{w}_1 &= \underset{\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, w_1}{\arg\max} \log\left(\prod_{i=1}^{a} p(x_i|y = 1)\right) \\ \hat{\mu_3}, \hat{\mu_4}, \hat{\sigma_3}^2, \hat{\sigma_4}^2, \hat{w}_2 &= \underset{\mu_3, \mu_4, \sigma_3^2, \sigma_4^2, w_2}{\arg\max} \log\left(\prod_{j=1}^{b} p(x_j|y = 0)\right). \end{aligned} \tag{3.17}$$

Pro Poissonovo rozdělení existuje však analytické řešení

$$\hat{\lambda} = \frac{1}{d} \sum_{j=1}^{d} x_j, \tag{3.18}$$

čímž je výběrový průměr hodnot. Není tedy potřeba maximalizovat věrohodnostní funkci Poissonova rozdělení numericky. Nutno podotknout, že složitější GM se pomocí MLE většinou neodhaduje. Pro odhad parametrů GM se běžně používá robustnější metoda, tzv. EM algoritmus [1] *(Expectation Maximization)*, který je iterativní, značně rychlejší a navíc využívá latentních proměnných. K odhadu tedy potřebujeme nějakou dodatečnou informaci, typicky informaci o tom, do kterého shluku (*clusteru*) daný bod patří. Nicméně, že zde funguje i MLE, se můžeme přesvědčit na obrázku 3.4.

V této chvíli bychom chtěli rozhodnout, do které třídy klient patří. Sestavíme následující dis-



(a) Distribuce $p(x|y=1)$

(b) Distribuce $p(x|y=0)$

Obrázek 3.4: Dvě GM, kde červenou a hnědou barvou jsou nakresleny skutečné distribuce, zeleně a žlutě jsou jejich MLE.

tribuce

$$\begin{aligned} p\left(\mathbf{x}, N_x^{(1)}|y=1\right) &= \left(\prod_{i=1}^{N_x^{(1)}} p(x_i|y=1)\right) \cdot p\left(N_x^{(1)}|y=1\right), \\ p\left(\mathbf{x}, N_x^{(0)}|y=0\right) &= \left(\prod_{i=1}^{N_x^{(0)}} p(x_i|y=0)\right) \cdot p\left(N_x^{(0)}|y=0\right) \end{aligned} \tag{3.19}$$

a s jejich pomocí provedeme test poměrem věrohodností [9] *(Likelihood Ratio Test, LRT)*

$$\begin{aligned} \Lambda_0(\mathbf{x}) &= \frac{p\left(\mathbf{x}, N_x^{(0)}|y=0\right)}{p\left(\mathbf{x}, N_x^{(1)}|y=1\right) + p\left(\mathbf{x}, N_x^{(0)}|y=0\right)}, \\ \Lambda_1(\mathbf{x}) &= \frac{p\left(\mathbf{x}, N_x^{(1)}|y=1\right)}{p\left(\mathbf{x}, N_x^{(1)}|y=1\right) + p\left(\mathbf{x}, N_x^{(0)}|y=0\right)}. \end{aligned} \tag{3.20}$$

První test udává pravděpodobnost s jakou jsou data vybraná z distribuce $p\left(x, N_x^{(0)}|y=0\right)$ a u druhého testu obdobně, pouze pro distribuci $p\left(x, N_x^{(1)}|y=1\right)$. Pokud tedy budeme mít $N_x^{(*)}$ pozorování z neznámé distribuce $p^*(x)$, jsme schopni rozhodnout do jaké třídy patří. V kontextu

finančního modelu nás klient žádá o půjčku a my na základě počtu a hodnoty jeho transakcí na jeho účtu chceme rozhodnout, zdali se jedná o člověka, který je schopen splatit potenciálně půjčené peníze, nebo nejedná. Stanovíme konstanty K_0 a K_1 takové, že

$$\begin{aligned} \Lambda_0(\mathbf{x}) &\leq K_0, \\ \Lambda_1(\mathbf{x}) &\leq K_1, \end{aligned} \tag{3.21}$$

které budou udávat pravděpodobnost, s jakou jsme ještě schopni přijmout hypotézu, jsou-li daná data vybraná z jednotlivých rozdělení $p(x, N_x^{(0)}|y = 0)$, $p(x, N_x^{(1)}|y = 1)$ nebo nejsou. Při malém počtu transakcí N_x nebude samozřejmě test přesný.

Aplikace

Pro demonstraci vygenerujeme data z následující GM

$$p(x^*) = 0.3 \cdot \mathcal{N}(-4.5, 1.5) + 0.7 \cdot \mathcal{N}(4.5, 1) \tag{3.22}$$

a jejich počet

$$p(N_x^*) = \mathrm{Po}(5). \tag{3.23}$$

K dispozici máme tedy hodnoty a počet transakcí nového klienta $\mathbf{x}^* = \left(x_1^*, \ldots, x_{N_x^*}^*\right)$. Pomocí MLE jsme odhadli všechny parametry

$$\begin{aligned} \boldsymbol{\theta} &= \left(\hat{\mu_1}, \hat{\mu_2}, \hat{\sigma_1}^2, \hat{\sigma_2}^2, \hat{\mu_3}, \hat{\mu_4}, \hat{\sigma_3}^2, \hat{\sigma_4}^2, \hat{\lambda_1}, \hat{\lambda_2}, \hat{w_1}, \hat{w_1}\right) \\ &= (-3, 3, 2, 1, -4, 5, 1.5, 1.5, 6, 5, 0.8, 0.4), \end{aligned} \tag{3.24}$$

které potřebujeme k provedení LRT testu. Jsou to odhadnuté parametry distribucí výše. Dále už jen dosadíme do (3.20) a obdržíme

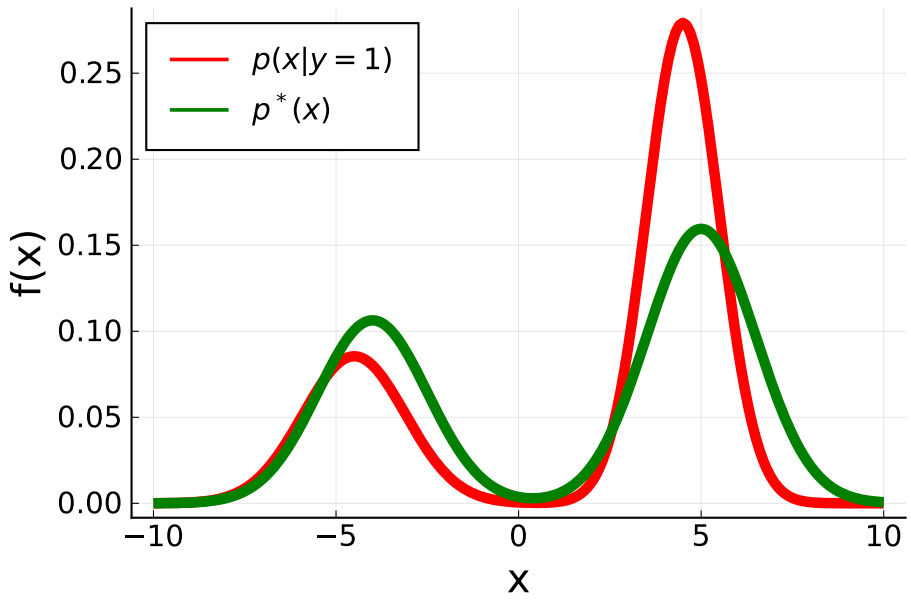
$$\begin{aligned} \Lambda_0(\mathbf{x}^*) &\doteq 0,08, \\ \Lambda_1(\mathbf{x}^*) &\doteq 0,92. \end{aligned} \tag{3.25}$$

Tímto jsme nového klienta zařadili do třídy *schopen splácet* a můžeme si dovolit mu poskytnout půjčku, test byl totiž dostatečně prokazatelný s $\Lambda_1(\mathbf{x}^*) \doteq 0,92$. Na obrázku 3.5, můžeme vidět podobnost distribuce $p(x|y = 1)$ a $p^*(x)$, je tedy zřejmé, že LRT by měl zařadit klienta do třídy $y = 1$.

Vylepšení

Položme si otázku, zdalipak nelze klasifikace do třídy *schopen splácet* nebo *není schopen splácet*, nějakým způsobem vylepšit. Může nastat situace, že bankovní záznam klienta neodpovídá ani jednomu modelu. Pak vzniká riziko, že LRT nebude přesný. Pro tento případ stanovíme například distribuci

$$p(x|y = 2) = \mathcal{N}\left(0, 10^5\right), \tag{3.26}$$



Obrázek 3.5: Rozdíl mezi distribucí $p(x|y = 1)$ a distribucí $p^*(x)$.

kde $y = 2$ bude indikátorem, že je něco s klientem v nepořádku. Jelikož budeme chtít opět testovat, jestli klient patří do této třídy, musíme opět sestrojit sdruženou distribuci

$$p(\mathbf{x}, N_x^{(2)}|y = 2) = \left(\prod_{i=1}^{N_x^{(2)}} p(x_i|y = 2) \right) \cdot p\left(N_x^{(2)}|y = 2\right). \tag{3.27}$$

Test poměrem věrohodností potom nabude tvaru

$$\Lambda_2(\mathbf{x}) = \frac{p(\mathbf{x}, N_x^{(2)}|y = 2)}{p(\mathbf{x}, N_x^{(2)}|y = 2) + p(\mathbf{x}, N_x^{(1)}|y = 1) + p(\mathbf{x}, N_x^{(0)}|y = 0)} \tag{3.28}$$

Můžeme ho nazvat jako test podivnosti klienta.

Závěr

V této práci bylo primárním cílem najít takovou distribuci, která umí generovat stromové struktury. Celou práci jsme na tento popud koncipovali do tří částí.

V první kapitole jsme se zaměřili na nezbytný matematický aparát, nutný při řešení této problematiky. Účelem této kapitoly bylo odvezení ELBO, které jsme poté předvedli na příkladu při hledání parametrů distribuce. Ve druhé kapitole jsme definovali generativní modely a uvedli, jak se liší od diskriminativních, přičemž jsme v dalším příkladě hledali generativní model daných dat. Dále jsme definovali neuronovou síť a představili jsme koncept metody variačního autoencoderu, kde jsme využili odvozené ELBO. Ve třetí kapitole jsme se zaměřili na stromové struktury a pokusili se na ně napasovat generativní model. Řešili jsme další příklad, kde jsme našli takovou distribuci, která umí jednoduché stromové struktury generovat, přičemž jsme využili multi–instančního učení a vnořeného prostoru. Ověřili jsme také, že na vyřešení takové úlohy lze za pomoci neuronové sítě využít i metoda variačního autoencoderu. V posledním příkladu jsme řešili zjednodušený finanční model popsaný pomocí Gaussovských směsí a rozhodovali jsme, do jaké třídy (schopen, či není schopen splácet půjčku) klient patří.

Motivací do budoucna je tuto práci rozšířit do takové míry, aby byla aplikovatelná na reálná data. Mělo by na ní být tudíž navázáno prací diplomovou, popřípadě dalším akademickým výzkumem.

Literatura

[1] BISHOP, Christopher M. : *Pattern recognition and machine learning*. [New York]: Springer, 2006. Information science and statistics. ISBN 0-387-31073-8.

[2] MANDLÍK, Šimon. *Mapování internetu - modelování interakcí entit v komplexních heterogenních sítích*. Praha, 2020. Diplomová práce. České vysoké učení technické v Praze. Výpočetní a informační centrum.

[3] JIROVSKÝ, Lukáš. *Teorie grafů ve výuce na střední škole*. Praha, 2008. Diplomová práce. Univerzita Karlova, Matematicko-fyzikální fakulta.

[4] PEVNÝ, Tomáš a Petr SOMOL. Discriminative models for multi-instance problems with tree structure. *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*. 2016, 83–91.

[5] SHAFKAT, Irhum. *Intuitively understanding variational autoencoders* [online]. [cit. 2020-07-22]. Dostupné z: `https://towardsdatascience.com/intuitivelyunderstanding-variational-autoencoders-1bfe67eb5daf`

[6] RUDER, Sebastian. An overview of gradient descent optimization algorithms. *ArXiv preprint arXiv:1609.04747*. 2016, 1–14.

[7] KINGMA, Diederik P. a Max WELLING. Auto-encoding variational bayes. ArXiv preprint arXiv:1312.6114. 2013, 1–14.

[8] KOVÁŘ, Jan a Niels VAN DE MEER. *Zápisky z míry a pravděpodobnosti.* Praha, 2020. Vysokoškolská skripta. Fakulta jaderná a fyzikálně inženýrská ČVUT v Praze.

[9] KŮS, Václav a Martin KOVANDA.*Matematická statistika.* Praha, 2020. Vysokoškolská skripta. Fakulta jaderná a fyzikálně inženýrská ČVUT v Praze.

[10] LEARNED-MILLER, Eric. *Vector, Matrix, and Tensor Derivatives* [online]. [cit. 2020-07-22]. Dostupné z: `http://cs231n.stanford.edu/vecDerivs.pdf`

[11] JEFFREYS, Harold. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences.* 1946, 1–9.

[12] JORDAN, Michael Irwin a Andrew Y NG. Advances in neural information processing systems: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. 2002.

[13] ŠMÍDL, Václav. *Linear Regression, Automatic Relevance Determination* [online]. Czech Academy of Sciences [cit. 2020-07-22]. Dostupné z: http://staff.utia.cas.cz/smidl/files/hbm2020/prezentace03_20.pdf

[14] COMMENGES, Daniel. Information Theory and Statistics: an overview. *ArXiv preprint arXiv:1511.00860.* 2015, 1–22.

[15] PEVNÝ, Tomáš a Marek DĚDIČ. Nested Multiple Instance Learning in Modelling of HTTP network traffic.*ArXiv preprint arXiv:2002.04059.* 2020, 1–13.

[16] BEZANSON, Jeff, Stefan KARPINSKI, Viral B. SHAH a Alan EDELMAN. Julia: A fast dynamic language for technical computing. *ArXiv preprint arXiv:1209.5145.* 2012, 1–27.

[17] YANG, Xitong. *Understanding the Variational Lower Bound* [online]. 2017 [cit. 2020-07-23]. Dostupné z: http://legacydirs.umiacs.umd.edu/~xyang35/files/understanding-variational-lower.pdf