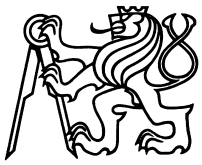
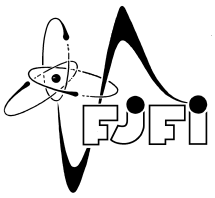


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



Metody strojového učení ve fyzice pevných látek

Methods of Machine Learning in Condensed Matter Physics

Bakalářská práce

Autor: **Jan Trödler**

Vedoucí práce: **doc. RNDr. Jan Vybíral, PhD.**

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:	Jan Trödler
Studijní program:	Aplikace přírodních věd
Obor:	Matematické inženýrství
Zaměření:	Aplikované matematicko-stochastické metody
Název práce (česky):	Metody strojového učení ve fyzice pevných látek
Název práce (anglicky):	Methods of machine learning in condensed matter physics

Pokyny pro vypracování:

1. Úkolem práce bude zpracování dat z oboru fyziky pevných látek popisující chování potenciálních materiálů určených pro výrobu solárních panelů. Student se seznámí se strukturou dat (chemické složení, geometrické parametry, formation energy, bandgap energy).
2. Student se seznámí se základními metodami strojového učení (LASSO, Kernel Ridge Regression) a metodami reprezentace (zejména geometrických) dat.
3. Student se pokusí vhodnou kombinací stávajících metod a různých reprezentací dat překonat současné algoritmy zpracování těchto dat.
4. Na základě poznatků získaných v předchozí části student zkusí navrhnout modifikace a vylepšení jak použitých algoritmů, tak i vhodné reprezentace vstupních dat. Klíčovým úkolem bude využít analytické závislosti materiálových vlastností na parametrech jednotlivých materiálů.

Doporučená literatura:

1. L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, M. Scheffler, Big data of materials science - Critical role of the descriptor. Phys. Rev. Lett. 114, 2015, 105503.

2. L. M. Ghiringhelli, J. Vybiral, E. Ahmetchik, R. Ouyang, S. V. Levchenko, C. Draxl, M. Scheffler, Learning physical descriptors for materials science by compressed sensing. New Journal of Physics 19, 2017, 023017.

3. K. R. Mueller, S. Mika, G. Ratsch, K. Tsuda, B. Schoelkopf, An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks, 12(2), 2001, 181-201.

Jméno a pracoviště vedoucího bakalářské práce:

doc. RNDr. Jan Vybíral, Ph.D.
Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze, Trojanova 13, 12000 Praha

Jméno a pracoviště konzultanta:

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

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská
-1-

.....
děkan

Poděkování:
Chtěl bych zde poděkovat především svému školiteli, panu docentu Janu Vybíralovi, za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé bakalářské práce.

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

V Praze dne 16. července 2020 Jan Trödler

Název práce:

Metody strojového učení ve fyzice pevných látek

Autor: Jan Trödler

Obor: Matematické inženýrství

Zaměření: Aplikované matematicko-stochastické metody

Druh práce: Bakalářská práce

Vedoucí práce: doc. RNDr. Jan Vybíral PhD., Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, ČVUT

Abstrakt: Strojové učení lze použít k efektivní předpovědi parametrů testovacích dat na základě dat trénovacích. Jedněmi z používaných metod strojového učení jsou metody Kernel Ridge Regression a LASSO, které obě vycházejí z lineární regrese. V této bakalářské práci bude čtenář nejprve seznámen s výše zmíněnými metodami na teoretické úrovni. Reálná aplikace metod probíhá na datech pocházejících z výzkumné oblasti zkoumající materiály potencionálně vhodné k výrobě solárních panelů. Vhodné vlastnosti těchto materiálů zavisí na dvou energiích, formační energii a energii zakázaného pásu. Všechny potřebné informace o konkrétních datech obdrží čtenář v druhé kapitole. Poslední částí této práce je vlastní program obsahující výpočty výstupních parametrů pomocí metody Kernel Ridge Regression a jeho postupné modifikace a vylepšení.

Klíčová slova: Kernel Ridge Regression, LASSO, strojové učení

Title:

Methods of Machine Learning in Condensed Matter Physics

Author: Jan Trödler

Abstract: Machine learning can be used to effectively predict test data parameters based on training data. One of the machine learning methods used is the Kernel Ridge Regression and LASSO methods, both of which are based on linear regression. In this bachelor's thesis, the reader will first be introduced to the above methods at a theoretical level. The real application of the methods is based on data from the research area examining materials potentially suitable for the production of solar panels. The suitable properties of these materials depend on two energies, the formation energy and the band gap energy. The reader will get all the necessary information about specific data in the second chapter. The last part of this work is a program containing calculations of output parameters using the Kernel Ridge Regression method and its gradual modifications and improvements.

Key words: Kernel Ridge Regression, LASSO, machine learning

Obsah

Úvod

Strojové učení značně mění podobu moderní doby. Usnadňuje a urychluje vývoj v širokém spektru odvětví. Metody strojového učení se snaží na základě trénovacích vzorků, u kterých jsou známé předpovídané hodnoty, předpovědět ty samé parametry pro data testovací. Ke strojovému učení patří například lineární regrese, metody využívající neuronové sítě, a tak dále. Využívanými metodami jsou také Kernel Ridge Regression a LASSO, kterým se věnuje tato bakalářská práce.

Obě zmíněné metody, tedy Kernel Ridge Regression i LASSO, vychází ze základní lineární regrese, což je metoda nejmenších čtverců. V první kapitole, která bude čistě teoretická, budou odvozeny všechny zmíněné teoretické problémy, tedy metoda nejmenších čtverců, Kernel Ridge Regression i LASSO.

Každá teoreticky odvozená metoda se vyvíjí za účelem praktické aplikace na reálný problém, tedy reálná data. Ani v této bakalařské práci není záměrem pouze teoretické seznámení se se zadaným tématem metod strojového učení, ale také vyzkoušení aplikace na konkrétních data. Pro tento účel byla vybrána soutěž NOMAD 2018, která probíhala na serveru Kaggle. S jakými daty soutěž probíhala a proč zrovna s těmito daty účastníci pracovali, to vše se čtenář této bakalářské práce dozví ve druhé kapitole, která pojednává právě o datech z již zmíněné soutěže.

Posledním úkolem, který vyplynul ze zadání této práce, je aplikovat výše zmíněné metody na data, o kterých je řeč v kapitole 2. Implementace této metody bude provedena v programovacím jazyce Matlab. Postupné zlepšování prováděných pokusů je detailně k nalezení v kapitole 3. Zde je čtenář podrobně seznámen se strukturou kódu jednotlivých programů, s tím, proč se daná část programu používá, k jakým výsledkům se dospělo a v neposlední řadě také se srovnáním dosažených výsledků s ostatními účastníky ze soutěže NOMAD 2018.

Tato bakalářská práce má tedy za cíl proniknout do problematiky zmíněných metod strojového učení, dále pak seznámit se s konkrétními daty, na kterých se strojové učení může provádět, a v poslední řadě je také žádoucí si vyzkoušet konkrétní implemetaci do programovacího jazyka (zde do jazyka Matlab) a spuštění programů.

Kapitola 1

Teorie

V první kapitole se zaměříme na čistě matematickou stránku našeho problému strojového učení, konkrétně tedy metody *Kernel Ridge Regression*. Avšak než se dostaneme k této konkrétní metodě, odvodíme si nejpve metodu nejmenších čtverců, jelikož Kernel Ridge Regression z ní vychází. Mezitím také stručně zmíníme druhou z metod, a to *LASSO*.

1.1 Metoda nejmenších čtverců

Metoda nejmenších čtverců je jednou z nejrozšířenějších lineárních metod využívaných k nalezení funkční závislosti mezi různými daty.

1.1.1 Odvození metody

Mějme tedy N vstupních dat $\boldsymbol{x^1}, ..., \boldsymbol{x^N} \in \mathbb{R}^n$ a k nim N příslušných hodnot $y_1, ..., y_N \in \mathbb{R}$. Naším úkolem je nyní nalézt linearní závislost, která by nejlépe aproximovala hodnoty $\boldsymbol{y}$. Hledáme proto takové nejlepší $\boldsymbol{\alpha} \in \mathbb{R}^n$, aby $y_j \approx \sum_{i=1}^{n} \alpha_i x_i^j$, tedy aby $y_j \approx \langle \boldsymbol{\alpha}, \boldsymbol{x^j} \rangle$ pro všechna j $\in \{1, ..., N\}$. Proto minimalizujeme

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \sum_{i=1}^{N} \left(y_i - \langle \boldsymbol{\alpha}, \boldsymbol{x^i} \rangle \right)^2 = \min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \| \boldsymbol{y} - X\boldsymbol{\alpha} \|_2^2, \tag{1.1}$$

kde jsme využili a budeme využívat následující značení:

- $\boldsymbol{y} \overset{ozn.}{=} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$ - vektory budou sloupcové
- X je matice, jejíž *i-tý* řádek je vektor $(\boldsymbol{x^i})^T$, tedy $X_{ij} = (\boldsymbol{x^i})_j$
- $f(\boldsymbol{\alpha}) = \| \boldsymbol{y} - X\boldsymbol{\alpha} \|_2^2$
- $\| \, . \, \|_2$ je klasická euklidovská norma; budeme ji odteď označovat pouze jako $\| \, . \, \|$.

Máme tedy nyní funkci $f(\boldsymbol{\alpha})$ a hledáme její minimum. Dokážeme, že tato funkce je konvexní na celém svém definičním oboru, a proto následně budeme s jistotou vědět, že její stacionární bod je tedy bodem globálního minima. Přesvědčme se tedy nejprve o konvexnosti funkce.

Definice 1.1. Funkce g spojitá na (a, b) je konvexní na tomto intervalu právě tehdy, když:

$$\forall \lambda \in (0,1);\ x, y \in (a,b),\ x < y:\ g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda)g(y).$$

Ověřme proto nyní konvexnost pro funkci $\sqrt{f(\boldsymbol{\alpha})}$:

$$\begin{aligned}\sqrt{f(\lambda\boldsymbol{\beta} + (1-\lambda)\boldsymbol{\gamma})} &= \|\boldsymbol{y} - X(\lambda\boldsymbol{\beta} + (1-\lambda)\boldsymbol{\gamma})\| = \|(\lambda\boldsymbol{y} + (1-\lambda)\boldsymbol{y}) - X(\lambda\boldsymbol{\beta} + (1-\lambda)\boldsymbol{\gamma})\| \\ &= \|\lambda(\boldsymbol{y} - X\boldsymbol{\beta}) + (1-\lambda)(\boldsymbol{y} - X\boldsymbol{\gamma})\| \overset{(1)}{\leq} \|\lambda(\boldsymbol{y} - X\boldsymbol{\beta})\| + \|(1-\lambda)(\boldsymbol{y} - X\boldsymbol{\gamma})\| \\ &= \lambda\|\boldsymbol{y} - X\boldsymbol{\beta}\| + (1-\lambda)\|\boldsymbol{y} - X\boldsymbol{\gamma}\| = \lambda\sqrt{f(\boldsymbol{\beta})} + (1-\lambda)\sqrt{f(\boldsymbol{\gamma})}.\end{aligned} \tag{1.2}$$

Tímto jsme dokázali konvexnost funkce $\sqrt{f(\boldsymbol{\alpha})}$, když jsme v bodě (1) použili trojúhelníkovou nerovnost. Za pomoci (1.2) ověříme nyní konvexnost funkce $f(\boldsymbol{\alpha})$:

$$\begin{aligned}f(\lambda\boldsymbol{\beta} + (1-\lambda)\boldsymbol{\gamma}) &\leq \lambda^2 f(\boldsymbol{\beta}) + 2\lambda(1-\lambda)\sqrt{f(\boldsymbol{\beta})}\sqrt{f(\boldsymbol{\gamma})} + (1-\lambda)^2 f(\boldsymbol{\gamma}) \\ &= \lambda f(\boldsymbol{\beta}) + \lambda(\lambda-1)f(\boldsymbol{\beta}) + (1-\lambda)f(\boldsymbol{\gamma}) + \lambda(\lambda-1)f(\boldsymbol{\gamma}) + 2\lambda(1-\lambda)\sqrt{f(\boldsymbol{\beta})f(\boldsymbol{\gamma})} \\ &= \lambda f(\boldsymbol{\beta}) + (1-\lambda)f(\boldsymbol{\gamma}) + \underbrace{\lambda(\lambda-1)}_{\leq 0}\underbrace{[f(\boldsymbol{\beta}) + f(\boldsymbol{\gamma}) - 2\sqrt{f(\boldsymbol{\beta})f(\boldsymbol{\gamma})}]}_{\geq 0} \\ &\leq \lambda f(\boldsymbol{\beta}) + (1-\lambda)f(\boldsymbol{\gamma}),\end{aligned}$$

z čehož vyplývá konvexnost fukce $f(\boldsymbol{\alpha})$. Nyní nám již tedy nic nebrání v tom, abychom nalezli extrém funkce $f(\boldsymbol{\alpha})$. To provedeme tak, že položíme $\nabla f = 0$ a vypočítáme odpovídající kořen. Proto tedy

$$\begin{aligned}\frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} &= 2\sum_{j=1}^{N}\Big(y_j - \sum_{i=1}^{n}\alpha_i(x^j)_i\Big)(-x^j)_k = -2\sum_{j=1}^{N}\Big(y_j - \langle\boldsymbol{\alpha}, \boldsymbol{x^j}\rangle\Big)(x^j)_k \\ &= -2\sum_{j=1}^{N} y_j(x^j)_k + 2\sum_{j=1}^{N}\Big(\langle\boldsymbol{\alpha}, \boldsymbol{x^j}\rangle\Big)(x^j)_k \\ &= -2\sum_{j=1}^{N}(X^T)_{k,j}(y)_j + 2\sum_{j=1}^{N}(X^T)_{k,j}(X\alpha)_j = -2(X^T y)_k + 2(X^T X\alpha)_k.\end{aligned} \tag{1.3}$$

Tudíž gradient

$$\nabla f(\boldsymbol{\alpha}) = -2(X^T\boldsymbol{y}) + 2(X^T X\boldsymbol{\alpha}),$$

který položíme roven nule a řešením této soustavy je

$$\boldsymbol{\alpha} = (X^T X)^{-1}X^T\boldsymbol{y}, \tag{1.4}$$

pokud existuje inverzní matice $(X^T X)^{-1}$.

1.1.2 Tichonovova regularizace

V předešlé části jsme si odvodili metodu nejmenších čtverců. Pro reálné využití se však tato obecná metoda vylepšuje pomocí tzv. Tichonovovy regularizace. Ta spočívá v tom, že do (1.1) přidáme člen, který penalizuje vektory $\boldsymbol{\alpha}$ s větší normou. Vše se dělá kvůli problému s tzv. *předeterminovaností systému*. Chceme proto získat takové $\boldsymbol{\alpha}$, které je nejmenší (ve smyslu normy).

Budeme tedy hledat extrém funkce

$$g(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha}) + \lambda\|\boldsymbol{\alpha}\|^2 = \|\,\boldsymbol{y} - X\boldsymbol{\alpha}\,\|^2 + \lambda\|\boldsymbol{\alpha}\|^2. \tag{1.5}$$

Opět bychom měli ověřit konvexnost této funkce, nicméně tato úloha bude velmi snadná, neboť funkce $f(\boldsymbol{\alpha})$ je konvexní dle ověření v části 1.1.1 a norma je zřejmě také konvexní. Stačí tedy najít řešení rovnice $\nabla g(\boldsymbol{\alpha}) = 0$. Vypočítejme parciální derivaci

$$\begin{aligned}\frac{\partial g(\boldsymbol{\alpha})}{\partial \alpha_k} &= \frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} + \frac{\partial(\lambda\|\boldsymbol{\alpha}\|^2)}{\partial \alpha_k} \\ &= -2(X^T\boldsymbol{y})_k + 2(X^TX\boldsymbol{\alpha})_k + 2\lambda\alpha_k,\end{aligned} \tag{1.6}$$

kde jsme využili již vypočítaného výsledku v (1.3). Celkem tedy dostáváme gradient

$$\nabla g(\boldsymbol{\alpha}) = -2(X^T\boldsymbol{y}) + 2(X^TX\boldsymbol{\alpha}) + 2\lambda\boldsymbol{\alpha}, \tag{1.7}$$

který opět položíme roven nule a řešení již získáváme ve tvaru

$$\boldsymbol{\alpha} = (X^TX + \lambda\mathbb{I})^{-1}X^T\boldsymbol{y}, \tag{1.8}$$

za předpokladu, že existuje $(X^TX + \lambda\mathbb{I})^{-1}$ (pozn.: značení $\mathbb{I}$ budeme používat pro identickou matici). Již jsme tedy obdrželi výsledek pro výpočet $\boldsymbol{\alpha}$ potřebný pro reguralizovanou metodu nejmenších čtverců.

1.2 LASSO

V této části se stručně podíváme na druhou zmiňovanou metodu strojového učení, která rovněž vychází z metody nejmenších čtverců. Jedná se o metodu **LASSO**, což je zkratka pocházející z anglického názvu *least absolute shrinkage and selection operator*. Metoda má za cíl najít takové $\boldsymbol{\alpha}$, které minimalizuje následující výraz

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \sum_{i=1}^{N} \left(y_i - \langle\boldsymbol{\alpha}, \boldsymbol{x}^i\rangle\right)^2 + \lambda \sum_{i=1}^{n} |\alpha_i| = \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{1.9}$$

kde $\lambda \geq 0$. Připomeneme si také pojem p-normy.

Definice 1.2. Nechť $\boldsymbol{x} \in \mathbb{R}^n$ a $p \geq 1$. Pak definujeme p-normu vektoru $\boldsymbol{x}$ jako

$$\|\boldsymbol{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}.$$

V našem případě tedy v (1.9) vystupuje p-norma vektoru $\boldsymbol{\alpha}$, přičemž za p volíme 1. Vidíme také, že úloha je velmi podobná úloze regularizované metody nejmenších čtverců odvozené v části 1.1.2. Rozdílem je pouze to, že v Tichonovově regularizaci z části 1.1.2 vystupuje v (1.5) norma vektoru $\boldsymbol{\alpha}$ tvaru $\|\boldsymbol{\alpha}\|_2$, v úloze metody LASSO je pak norma tvaru $\|\boldsymbol{\alpha}\|_1$. Proto také LASSO zmiňujeme na tomto místě, hned po regularizované mětodě nejmenších čtverců.

Bohužel na rozdíl od úlohy z Tichonovovy regularizace neexistuje pro výraz (1.9) analytické řešení. Pro výpočet $\boldsymbol{\alpha}$ se proto používají numerické metody. Nicméně pokud takové $\boldsymbol{\alpha}$ numericky najdeme, pokračujeme dále již totožně jako v části 1.1.1.

1.3 Kernel Ridge Regression

V předchozí části jsme odvodili metodu nejmenších čtverců. Tato metoda funguje spolehlivě, nicméně má jednu velikou limitaci. V předpisu (1.8) pro $\boldsymbol{\alpha}$ vidíme, že k tomu, abychom zjistili konkrétní $\boldsymbol{\alpha}$, potřebujeme počítat inverzní matici $(X^TX + \lambda\mathbb{I})^{-1}$, kde vystupuje součin X^TX. Proč zavedeme metodu Kernel Ridge Regression, která využívá tzv. *Kernel triku*, bude objasněno v této části. K metodě samotné dospějeme v několika krocích. Více podrobností o tomto problému je k nalezení v části 1.3.3.

1.3.1 Motivace

Podívejme se nejprve na člen X^TX v předpisu (1.8). V sekci 1.1.1 jsme zavedli jisté značení. Z toho vyplývá, že matice X^TX vypadá takto (v souladu s již zavedeným značením, tedy že X je matice, jejíž *j-tý* řádek je vektor $(\boldsymbol{x^j})^T$)

$$(X^TX)_{ij} = \sum_{k=1}^{N} (x_i^k)(x_j^k). \tag{1.10}$$

Vidíme, že toto nám připomíná skalární součin, avšak skalární součin to není. V sumě mezi sebou násobíme vždy stejné složky jiných vektorů. Zkusme se tedy podívat na matici, kde matice v (1.10) vynásobíme v opačném pořadí. Získáváme

$$(XX^T)_{ij} = \sum_{k=1}^{n} (x_k^i)(x_k^j), \tag{1.11}$$

což už však je již zmíněný skalární součin mezi vektory $\boldsymbol{x^i}$ a $\boldsymbol{x^j}$, tedy

$$(XX^T)_{ij} = \langle \boldsymbol{x^i}, \boldsymbol{x^j} \rangle. \tag{1.12}$$

Nyní se budeme snažit přepsat (1.8) pomocí výrazu, který by obsahoval místo X^TX pouze členy XX^T. A právě k tomu využijeme následující identitu.

1.3.2 Pomocná identita

Lemma 1.3. *Nechť* $\mathbb{I},\ A,\ B,\ C,\ D \in \mathbb{R}^{n,n}$ *a existují matice* $A^{-1},\ \left(A + BCD\right)^{-1},\ \left(\mathbb{I} + CDA^{-1}B\right)^{-1}$. *Pak*

$$\left(A + BCD\right)^{-1} BC = A^{-1}B\left(\mathbb{I} + CDA^{-1}B\right)^{-1} C. \tag{1.13}$$

Důkaz. Nejprve si odvodíme jednoduchý vztah

$$P(\mathbb{I} + QP) = (\mathbb{I} + PQ)P \implies (\mathbb{I} + PQ)^{-1}P = P(\mathbb{I} + QP)^{-1}, \tag{1.14}$$

který nyní aplikujeme na levou stranu v (1.13)

$$\left(A + BCD\right)^{-1} BC = A^{-1} \underbrace{\left(\mathbb{I} + BCDA^{-1}\right)B}_{\text{využijeme (1.14)}} C = A^{-1}B\left(\mathbb{I} + CDA^{-1}B\right)^{-1} C. \tag{1.15}$$

□

Toto lemma lze upravit také pro obdélníkové matice, vždy si však musí odpovídat rozměry matic (nutno zaručit možnost násobení matic mezi sebou) a také musí existovat potřebné inverzní matice.

1.3.3 Odvození metody

Pro samotné odvození využijeme již získaný výsledek z části 1.1. Nejprve však provedeme přechod od vektoru $\boldsymbol{x}$ k vektoru $\boldsymbol{\phi(x)}$.

Tento přechod je zásadní v celé teorii metody Kernel Ridge Regression. V pouhé lineární regresi (metoda nejmenších čtverců) nastává problém v případě, že je dat mnoho a jsou shluklé u sebe. Neumíme pak vytvořit přesnou předpověď pro nové hodnoty. Respektive umíme, nicméně předpověď bývá nepřesná. Když však přejdeme nějakým způsobem do prostoru s vyšší dimenzí, data se po prostoru více

„rozprostřou" (oddělí se od sebe) a my budeme také schopni najít předpis pro $\boldsymbol{\alpha}$ z rovnosti (1.8). Zároveň však toto $\boldsymbol{\alpha}$ bude mnohem přesněji aproximovat lineární závislost mezi těmito daty. Nelinearitu mezi daty v původním prostoru s původní dimenzí překonáme zobrazením do nového prostoru s vyšší dimenzí a nalezneme lineární fit.

Vezměme funkci

$$\phi : \mathbb{R}^n \to \mathbb{R}^D,$$

která tedy každému vektoru $\boldsymbol{x}$ přiřazuje vektor $\boldsymbol{\phi}(\boldsymbol{x})$ (tj. $\boldsymbol{x} \mapsto \boldsymbol{\phi}(\boldsymbol{x}) \ \ \forall \boldsymbol{x} \in \mathbb{R}^n$). Nutkno podotknout, že dimenze D bývá vždy větší než původní n, a to mnohonásobně. Dokonce ve většině případů je $D = +\infty$.

Vytvořme nyní novou matici Φ

$$\Phi \overset{ozn.}{=} \phi(X) = \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x^1})^T \\ \boldsymbol{\phi}(\boldsymbol{x^2})^T \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x^N})^T \end{pmatrix} \tag{1.16}$$

a dosadíme do (1.8) místo X

$$\hat{\boldsymbol{\alpha}} = (\Phi^T\Phi + \lambda\mathbb{I})^{-1}\Phi^T\boldsymbol{y}. \tag{1.17}$$

Nyní využijeme lemma 1.3, kde za jednotlivé matice v (1.13) bereme

$$A = \mathbb{I},\ B = \frac{1}{\lambda}\Phi^T,\ C = \mathbb{I},\ D = \Phi \tag{1.18}$$

a díky tomu přepíšeme vztah (1.17)

$$\hat{\boldsymbol{\alpha}} = (\Phi^T\Phi + \lambda\mathbb{I})^{-1}\Phi^T\boldsymbol{y} = \Phi^T \underbrace{(\Phi\Phi^T + \lambda\mathbb{I})^{-1}\boldsymbol{y}}_{\beta}. \tag{1.19}$$

V tomto vztahu nám vystupuje člen $\Phi\Phi^T$, který, jak již bylo zmíněno v části 1.3.1, tvoří matici, jejíž *ij-tý* prvek je skalární součin vektorů $\boldsymbol{\phi}(\boldsymbol{x^i})$ a $\boldsymbol{\phi}(\boldsymbol{x^j})$. Pro tuto matici zavedeme nové označení, a to K. Tedy

$$K_{ij} = \langle \boldsymbol{\phi}(\boldsymbol{x^i}), \boldsymbol{\phi}(\boldsymbol{x^j}) \rangle =: k(\boldsymbol{x^i}, \boldsymbol{x^j}). \tag{1.20}$$

Tomuto zobrazení $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ říkáme ***jádro***.

1.3.4 Kernel trik

V předchozí části jsme dospěli do nejdůležitější části metody Kernel Ridge Regression. Ve vztahu (1.20) jsme definovali jistým předpisem **jádro** k, které působí na vektory $\boldsymbol{x^i}$ a $\boldsymbol{x^j}$ jako

$$k(\boldsymbol{x^i}, \boldsymbol{x^j}) = \langle \boldsymbol{\phi}(\boldsymbol{x^i}), \boldsymbol{\phi}(\boldsymbol{x^j}) \rangle.$$

Nyní se můžeme ptát, proč celý postup vedeme tímto směrem. Odpověd nám dává *tzv.* ***Mercerova věta***.

Věta 1.4 (Mercerova věta)**.** *Nechť $k : C \times C \to \mathbb{R}$ je spojité jádro pozitivně-definitního integrálního operátoru na Hilbertově prostoru $\mathcal{L}_2(C)$, kde $C \subset \mathbb{R}^N$, tzn. pro každou funkci $f \in \mathcal{L}_2(C)$*

$$\int_C k(x, y) f(x) f(y)\, \mathrm{d}x\, \mathrm{d}y \geq 0.$$

Pak existuje prostor $\mathcal{F}$ a zobrazení $\phi : \mathbb{R}^N \to \mathcal{F}$ takové, že $k(x, y) = \langle \phi(x), \phi(y) \rangle$.

Znění věty vychází z informací na str. 140 v [14]. Toto tvrzení je nesmírně přínosné. Zaručuje nám, že pokud vezmeme vhodné k vyhovující podmínce v předchozím tvrzení, tedy například $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2}\right)$, jistě bude existovat zobrazení ϕ rovněž z předchozího tvrzení. Překvapivé je, že dokonce ani nemusíme znát předpis tohoto ϕ. Ve většině případů opravdu tento konkrétní předpis neznáme. Je to proto, že u spojitého jádra k je $dim\mathcal{F} = +\infty$. Nám však stačí, že Mercerova věta zaručuje existenci.

Vraťme se nyní úplně zpět k (1.19). Dospěli jsme k tomu, že $\hat{\boldsymbol{\alpha}}$ můžeme vyjádřit jako

$$\hat{\boldsymbol{\alpha}} = \Phi^T\beta = \sum_{i=1}^{N}\beta_i\boldsymbol{\phi}(\boldsymbol{x^i}) = \Phi^T(\Phi\Phi^T + \lambda\mathbb{I})^{-1}\boldsymbol{y} = \Phi^T\Big(K + \lambda\mathbb{I}\Big)^{-1}\boldsymbol{y}. \tag{1.21}$$

Vše samozřejmě směřujeme k důvodu, proč zde celou metodu zavádíme. Chtěli jsme, abychom aproximovali závislosti mezi vstupními daty a výstupními hodnotami. Když tedy nyní dostaneme nový vstup x a budeme k němu chtít předpovědět (tj. dopočítat) hodnotu parametru y', provedeme

$$y' \approx \langle\hat{\boldsymbol{\alpha}}, \boldsymbol{\phi}(\boldsymbol{x})\rangle = \hat{\boldsymbol{\alpha}}^T\boldsymbol{\phi}(\boldsymbol{x}) = \underbrace{\boldsymbol{y}^T\Big(K + \lambda\mathbb{I}\Big)^{-1}}_{\Sigma}\underbrace{\Phi\boldsymbol{\phi}(\boldsymbol{x})}_{\boldsymbol{\kappa}(\boldsymbol{x})}. \tag{1.22}$$

Rozepišme si nyní $\boldsymbol{\kappa}(\boldsymbol{x})$ za pomoci doposud zavedeného značení

$$\boldsymbol{\kappa}(\boldsymbol{x}) = \begin{pmatrix}\boldsymbol{\phi}(\boldsymbol{x^1})^T\\ \boldsymbol{\phi}(\boldsymbol{x^2})^T\\ \vdots\\ \boldsymbol{\phi}(\boldsymbol{x^N})^T\end{pmatrix}\cdot\Big(\boldsymbol{\phi}(\boldsymbol{x})\Big) = \begin{pmatrix}\langle\boldsymbol{\phi}(\boldsymbol{x^1}), \boldsymbol{\phi}(\boldsymbol{x})\rangle\\ \langle\boldsymbol{\phi}(\boldsymbol{x^2}), \boldsymbol{\phi}(\boldsymbol{x})\rangle\\ \vdots\\ \langle\boldsymbol{\phi}(\boldsymbol{x^N}), \boldsymbol{\phi}(\boldsymbol{x})\rangle\end{pmatrix} = \begin{pmatrix}k(\boldsymbol{x^1}, \boldsymbol{x})\\ k(\boldsymbol{x^2}, \boldsymbol{x})\\ \vdots\\ k(\boldsymbol{x^N}, \boldsymbol{x})\end{pmatrix}. \tag{1.23}$$

Je zřejmé, že vektor $\boldsymbol{\kappa}(\boldsymbol{x})$ je sestaven ze složek, které se dají vypočítat pouze ze znalosti jádra k. Závěrem tedy přepíšeme vztah (1.22)

$$y' \approx \boldsymbol{y}^T\Big(K + \lambda\mathbb{I}\Big)^{-1}\boldsymbol{\kappa}(\boldsymbol{x}). \tag{1.24}$$

Jak můžeme vidět, ve výsledném vztahu vystupují pouze data $\boldsymbol{y}$, matice K, parametr λ, identická matice a vektor $\boldsymbol{\kappa}(\boldsymbol{x})$. Přitom složky matice K a vektoru $\boldsymbol{\kappa}(\boldsymbol{x})$ jsme schopni napočítat jen a pouze pomocí jádra k, neboť $K_{ij} = k(\boldsymbol{x^i}, \boldsymbol{x^j})$ a $(\boldsymbol{\kappa}(\boldsymbol{x}))_i = k(\boldsymbol{x^i}, \boldsymbol{x})$. Vše jsme tedy schopni (resp. výpočetní technika je schopna) napočítat velmi snadně a rychle. Toto byl také náš cíl. Zavést metodu, která bude efektivně předpovídat výsledky pro nová data.

1.4 Kernel Ridge Regression - jiné odvození

V této části zkusíme odvodit již odvozený vztah (1.24) druhým možným způsobem, a to pomocí Lagrangeových multiplikátorů.

Naším cílem je tedy minimalizovat

$$L_p = \sum_{i=1}^{N}\xi_i^2 \tag{1.25}$$

za podmínek

$$y_i - \boldsymbol{\alpha}^T\boldsymbol{\phi}(\boldsymbol{x^i}) = \xi_i \quad \forall i \in \{1, ..., N\} \tag{1.26}$$

$$\|\boldsymbol{\alpha}\| = B, \quad B \in \mathbb{R}. \tag{1.27}$$

Proč nyní uvažujeme podmínky (1.26) a (1.27), na to nám poskytne odpověď následující věta.

Věta 1.5. *Nechť* $g(\boldsymbol{\alpha}) = \|\boldsymbol{y} - X\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|^2$ *je funkce z* (1.5)*, kterou chceme minimalizovat (tj. hledat její extrém), přičemž* λ *je pevný parametr. Pak pro každé* λ *existuje* B *pevné tak, že argument minima* $g(\boldsymbol{\alpha})$ *je roven argumentu minima funkce* $f(\boldsymbol{\alpha})$*, kde funkce* $f(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \xi_i^2$ *za podmínek*

$$\xi_i = y_i - \boldsymbol{\alpha}^T \boldsymbol{x^i} \quad \forall i \in \{1, ..., N\} \tag{1.28}$$

$$\|\boldsymbol{\alpha}\| = B, \quad B \in (0, +\infty). \tag{1.29}$$

Důkaz. Vezměme si tedy konkrétní λ pevné a minimalizujme funkci $g(\boldsymbol{\alpha})$. Označme

$$\arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} g(\boldsymbol{\alpha}) = \boldsymbol{\alpha_0}.$$

Zřejmě pak platí, že

$$g(\boldsymbol{\alpha_0}) \leq g(\boldsymbol{\alpha}) \quad \forall \boldsymbol{\alpha} \in \mathbb{R}^n.$$

Za B tedy volíme $\|\boldsymbol{\alpha_0}\| = B$. Argument minima funkce $f(\boldsymbol{\alpha})$ je při volbě tohoto B roven také α_0.
Tedy

$$\arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} g(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} f(\boldsymbol{\alpha}) = \boldsymbol{\alpha_0}.$$

□

Poznámka 1.6. Věta 1.5 je vyslovená ve formě implikace. Tuto větu lze vyslovit i dokázat ve formě ekvivalence. Nicméně důkaz je velmi obtížný a pro naše účely stačí tato jednostranná implikace.

Již byl tedy zaveden přechod od $\boldsymbol{x}$ k vektoru $\boldsymbol{\phi(x)}$ komentovaný v části 1.3.3. Zavedeme Lagrangeovy multiplikátory $(\beta_i)_{i=1}^N$ a λ a sestavíme Lagrangián, který bude mít tvar

$$L\Big((\xi_i)_{i=1}^N, (\beta_i)_{i=1}^N, \lambda, \boldsymbol{\alpha}\Big) = \sum_{i=1}^{N} \xi_i^2 + \sum_{i=1}^{N} \beta_i \Big[y_i - \boldsymbol{\alpha}^T \boldsymbol{\phi}(\boldsymbol{x^i}) - \xi_i\Big] + \lambda(\|\boldsymbol{\alpha}\|^2 - B^2). \tag{1.30}$$

Nyní se budeme snažit naléz extrém tohoto Lagrangiánu. Z teorie víme, že extrém odpovídá řešení úlohy dle *Karush–Kuhn–Tuckerových podmínek*

$$\frac{\partial L}{\partial \xi_i} = 0 \ \ \& \ \ \frac{\partial L}{\partial \alpha_i} = 0 \quad \forall i \in \{1, ..., N\}.$$

Provedeme tedy výpočet parciálních derivací, které položíme rovny nule

$$\frac{\partial L}{\partial \xi_i} = 2\xi_i - \beta_i = 0 \implies 2\xi_i = \beta_i \quad \forall i \in \{1, ..., N\} \tag{1.31}$$

$$\frac{\partial L}{\partial \alpha_k} = -\sum_{i=1}^{N} \beta_i \Big(\boldsymbol{\phi}(\boldsymbol{x^i})\Big)_k + 2\lambda\alpha_k \tag{1.32}$$

$$-\sum_{i=1}^{N} \beta_i \boldsymbol{\phi}(\boldsymbol{x^i}) + 2\lambda\boldsymbol{\alpha} = \boldsymbol{0} \implies \boldsymbol{\alpha} = \frac{1}{2\lambda} \sum_{i=1}^{N} \beta_i \boldsymbol{\phi}(\boldsymbol{x^i}), \tag{1.33}$$

a získané výsledky dosadíme zpět do (1.30), čímž získáme Lagrangián v nových proměnných

$$\begin{aligned} L_D &= \sum_{i=1}^{N}\Big(\frac{\beta_i}{2}\Big)^2 + \sum_{i=1}^{N}\beta_i y_i - \sum_{i=1}^{N}\beta_i\frac{\beta_i}{2} - \Big(\sum_{i=1}^{N}\beta_i\boldsymbol{\phi}(\boldsymbol{x^i})\Big)\cdot\underbrace{\Big(\sum_{j=1}^{N}\frac{1}{2\lambda}\beta_j\boldsymbol{\phi}(\boldsymbol{x^j})\Big)}_{\boldsymbol{\alpha}\text{ z (1.33)}} + \lambda\frac{1}{4\lambda^2}\Big(\sum_{i=1}^{N}\beta_i\boldsymbol{\phi}(\boldsymbol{x^i})\Big)^2 - \lambda B^2 \\ &= \sum_{i=1}^{N}\Big(-\frac{\beta_i^2}{4} + \beta_i y_i\Big) - \frac{1}{4\lambda}\sum_{i=1}^{N}\sum_{j=1}^{N}\Big(\beta_i\beta_j\underbrace{\langle\boldsymbol{\phi}(\boldsymbol{x^i}),\boldsymbol{\phi}(\boldsymbol{x^j})\rangle}_{K_{ij}}\Big) - \lambda B^2, \end{aligned} \tag{1.34}$$

přičemž matice K je zachována dle značení z (1.20).

Nyní si předefinujeme nové proměnné $\alpha_i = \frac{\beta_i}{2\lambda}$, ve kterých přepíšeme Lagrangián L_D

$$\begin{aligned} L_D\Big((\alpha_i)_{i=1}^N\Big) &= -\lambda^2\sum_{i=1}^{N}\alpha_i^2 + 2\lambda\sum_{i=1}^{N}\alpha_i y_i - \lambda\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j K_{ij} - \lambda B^2 \\ &= -\lambda^2\|\boldsymbol{\alpha}\|^2 + 2\lambda\boldsymbol{\alpha}^T\boldsymbol{y} - \lambda\boldsymbol{\alpha}^T K\boldsymbol{\alpha} - \lambda B^2 \end{aligned} \tag{1.35}$$

a ten optimalizujeme (tj. minimalizujeme) přes všechny $\boldsymbol{\alpha}$ z $\mathbb{R}^n$. Opět budeme počítat parciální derivace, které položíme rovny nule, a získáme řešení pro $\boldsymbol{\alpha}$

$$\frac{\partial L_D}{\partial\alpha_i} = -2\lambda^2\alpha_i + 2\lambda y_i - 2\lambda\sum_{j=1}^{N}\alpha_j K_{ij} \tag{1.36}$$

$$\begin{aligned} -2\lambda^2\mathbb{I}\boldsymbol{\alpha} + 2\lambda\boldsymbol{y} - 2\lambda K\boldsymbol{\alpha} &= 0 \\ (\lambda\mathbb{I} + K)\boldsymbol{\alpha} &= \boldsymbol{y} \\ \boldsymbol{\alpha} &= (\lambda\mathbb{I} + K)^{-1}\boldsymbol{y}. \end{aligned} \tag{1.37}$$

Již jsme obdrželi předpis pro $\boldsymbol{\alpha}$.

1.5 Jádra metody Kernel Ridge Regression

Již víme, že se v metodě Kernel Ridge Regression ve výpočtu používá **jádro**. V této práci bylo použito *Gaussovo jádro*, nicméně není jediné, proto se v této krátké části podíváme i na použití dalších jader.

1.5.1 Podmínka

Abychom funkci $f : C \times C \to \mathcal{F}$ mohli označit za jádro použitelné pro metodu Kernel Ridge Regression (chápeme spojité), musí tato funkce splňovat předpoklady Mercerovy věty 1.4. Jádra mohou být například lineární, nelineární, exponenciální, polynomická či sigmoidní.

1.5.2 Příklady

V následující tabulce (2.1) jsou uvedeny příklady nejčastěji používaných jader. Jednotlivé konstanty $\theta, c, \gamma \in \mathbb{R}$, $d \in \mathbb{N}$ a $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $n \in \mathbb{N}$.

Tangens-hyperbolické jádro je hojně využíváno u neuronových sítí. Gaussovo a Laplaceovo jádro se využívá nejčastěji, jelikož dobře funguje na souborech, kde není známá žádná potenciální závislost dat.

Tabulka 1.1: Příklady jader

Název jádra	předpis pro $k(\boldsymbol{x}, \boldsymbol{y})$
Polynomické	$(\langle \boldsymbol{x}, \boldsymbol{y} \rangle + \theta)^d$
Gaussovo	$\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2c}\right)$
Laplaceovo	$\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|}{c}\right)$
Tangens-hyperbolické	$\tanh(\kappa\langle \boldsymbol{x}, \boldsymbol{y} \rangle + \gamma)$

1.5.3 Gaussovo jádro

Celá teorie Kernel Ridge Regression předpokládá, že zobrazení ϕ z věty 1.4 existuje. Nepotřebujeme znát však jeho přesný předpis. Nicméně pro ilustraci si zde ukážeme, jak by vypadalo toto ϕ pro jedno konkrétní jádro, a to námi používané Gaussovo jádro.

Naším úkolem je tedy najít takové ϕ, aby

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2c}\right) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle. \tag{1.38}$$

Za konstantu c hned volíme číslo 1, jelikož i v naší práci používáme jádro tvaru $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2}\right)$.

Nyní budeme upravovat výraz v (1.38)

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}) &= \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2}\right) = \exp\left(-\frac{1}{2}(\|\boldsymbol{x}\|^2 - 2\langle \boldsymbol{x}, \boldsymbol{y} \rangle + \|\boldsymbol{y}\|^2)\right) \\ &= \underbrace{\exp\left(-\frac{\|\boldsymbol{x}\|^2}{2}\right)}_{\alpha} \underbrace{\exp\left(\langle \boldsymbol{x}, \boldsymbol{y} \rangle\right)}_{\beta} \underbrace{\exp\left(-\frac{\|\boldsymbol{y}\|^2}{2}\right)}_{\gamma}. \end{aligned} \tag{1.39}$$

Vidíme, že v (1.39) výraz α závisí pouze na $\boldsymbol{x}$, naopak výraz γ pouze na $\boldsymbol{y}$. Výraz β si znovu rozepíšeme, tentokrát pomocí teorie Taylorova rozvoje exponenciální funkce

$$\begin{aligned} \beta &= \exp\left(\langle \boldsymbol{x}, \boldsymbol{y} \rangle\right) = 1 + \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{1!} + \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2}{2!} + \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle^3}{3!} + \dots \\ &= 1 + \frac{x_1 y_1 + x_2 y_2 + \dots + x_n y_n}{1!} + \frac{\left(x_1 y_1 + x_2 y_2 + \dots + x_n y_n\right)^2}{2!} + \dots \\ &= \left(1, \tfrac{1}{\sqrt{1!}} x_1, \tfrac{1}{\sqrt{1!}} x_2, \dots, \tfrac{1}{\sqrt{1!}} x_n, \tfrac{1}{\sqrt{2!}} x_1^2, \tfrac{1}{\sqrt{2!}} x_2^2, \dots, \tfrac{1}{\sqrt{2!}} x_n^2, \tfrac{1}{\sqrt{2!}} x_1 x_2, \dots\right) \cdot \begin{pmatrix} 1 \\ \frac{1}{\sqrt{1!}} y_1 \\ \frac{1}{\sqrt{1!}} y_2 \\ \vdots \\ \frac{1}{\sqrt{1!}} y_n \\ \frac{1}{\sqrt{2!}} y_1^2 \\ \frac{1}{\sqrt{2!}} y_2^2 \\ \vdots \\ \frac{1}{\sqrt{2!}} y_n^2 \\ \frac{1}{\sqrt{2!}} y_1 y_2 \\ \vdots \end{pmatrix}. \end{aligned} \tag{1.40}$$

Výraz β jsme rozepsali jako skalární součin dvou vektorů, kdy jeden z nich závisí pouze na vektoru $\boldsymbol{x}$ a druhý pouze na $\boldsymbol{y}$. Celkem tedy dostáváme, že předpis pro zobrazení ϕ pro Gaussovo jádro s parametrem $c = 1$ je

$$\phi(\boldsymbol{x}) = \exp\Big(-\frac{||\boldsymbol{x}||^2}{2}\Big)\Big(1, \tfrac{1}{\sqrt{1!}}x_1, \tfrac{1}{\sqrt{1!}}x_2, \ldots, \tfrac{1}{\sqrt{1!}}x_n, \tfrac{1}{\sqrt{2!}}x_1^2, \tfrac{1}{\sqrt{2!}}x_2^2, \ldots, \tfrac{1}{\sqrt{2!}}x_n^2, \tfrac{1}{\sqrt{2!}}x_1x_2, \ldots\Big)^T. \tag{1.41}$$

1.5.4 Ověření podmínky Mercerovy věty

V této části provedeme ověření podmínky z věty 1.4 zmíněné v části 1.5.1 pro některá jádra uvedená v tabulce 1.1. K tomu využijeme Bochnerovu větu. Nejprve však uvedeme ještě definici pozitivně definitní funkce.

Definice 1.7. O reálné funkci $\Phi : \mathbb{R}^n \to \mathbb{R}$ řekneme, že je pozitivně definitní, pokud matice $(\Phi(\boldsymbol{x_i} - \boldsymbol{x_j}))_{i,j=1}^N$ je pozitivně semidefinitní pro každé N z $\mathbb{N}$ a každou volbu $\boldsymbol{x_1}, \ldots, \boldsymbol{x_N}$ z $\mathbb{R}^n$.

Poznámka 1.8. Matice $(\Phi(\boldsymbol{x_i} - \boldsymbol{x_j}))_{i,j=1}^N$ je pozitivně semidefinitní právě tehdy, když

$$\sum_{i,j=1}^{N} a_i a_j \Phi(\boldsymbol{x_i} - \boldsymbol{x_j}) \geq 0 \quad \forall \boldsymbol{x_1}, \ldots, \boldsymbol{x_N} \in \mathbb{R}^n,\ \forall (a_1, \ldots, a_N) \in \mathbb{R}^N. \tag{1.42}$$

Věta 1.9 (Bochnerova věta)**.** *Nechť funkce $f(\boldsymbol{x})$ je spojitá na $\mathbb{R}^n$. Pak $f(\boldsymbol{x})$ je pozitivně definitní právě tehdy, když existuje funkce $g(\boldsymbol{x})$ taková, že:*

1. $g(\boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x} \in \mathbb{R}^n$, *a*
2. $f(\boldsymbol{x}) = \int\limits_{\mathbb{R}^n} g(\boldsymbol{\xi}) \exp(\imath \boldsymbol{\xi} \boldsymbol{x})\, \mathrm{d}\boldsymbol{\xi}$.

Poznámka 1.10. Znění věty 1.9 a více informací o ní lze nalézt na str. 19 v [10]. Funkce $f(\boldsymbol{x})$ z předchozí věty je tedy inverzní Fourierovou transformací funkce $g(\boldsymbol{x})$. Jinými slovy, pokud existuje funkce $g(\boldsymbol{x})$ taková, že je nezáporná na celém svém definičním oboru a funkce $f(\boldsymbol{x})$ je její inverzní Fourierovou transtormací, pak $k(\boldsymbol{x}, \boldsymbol{y})$ je jádro vyhovující Mercerově větě, pokud $k(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x} - \boldsymbol{y})$.

Příklad 1.11. Ověření provedeme pro jádro $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\Big(-\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{2}\Big)$. Nechť $\boldsymbol{x}$ a $\boldsymbol{y}$ jsou z $\mathbb{R}^n$. Potom máme ověřit, že

$$\int\limits_{C \times C} \exp\Big(-\frac{||\boldsymbol{x} - \boldsymbol{y}||^2}{2}\Big) f(\boldsymbol{x}) f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} \geq 0 \quad \forall C \subset \mathbb{R}^n.$$

Díky větě 1.9 a poznámce 1.10 víme, že stačí nalézt funkci $g(\boldsymbol{x})$, která je nezáporná a její Fourierova transformace je právě funkce $f(\boldsymbol{x}) = \exp\Big(-\frac{||\boldsymbol{x}||^2}{2}\Big)$. Z vlastností Fourierovy transformace fukce $f(\boldsymbol{x})$ víme, že

$$\hat{f}(\boldsymbol{x}) = \int\limits_{\mathbb{R}^n} \exp\Big(-\frac{||\boldsymbol{\xi}||^2}{2}\Big) \exp\Big(\imath \boldsymbol{\xi} \boldsymbol{x}\Big)\, \mathrm{d}\boldsymbol{\xi} = (2\pi)^{\frac{n}{2}} \exp\Big(-\frac{||\boldsymbol{x}||^2}{2}\Big). \tag{1.43}$$

Když tedy za $g(\boldsymbol{x})$ zvolíme

$$g(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}} f(\boldsymbol{x}) = (2\pi)^{-\frac{n}{2}} \exp\Big(-\frac{||\boldsymbol{x}||^2}{2}\Big), \tag{1.44}$$

která je jistě nezáporná, neboť $(2\pi)^{-\frac{n}{2}} \geq 0$ a funkce $\exp\Big(-\frac{||\boldsymbol{x}||^2}{2}\Big)$ je taktéž nezáporná na celém svém definičním oboru, pak její Fourierovou transformací je právě funkce $f(\boldsymbol{x})$. Proto můžeme potvrdit, že Gaussovo jádro $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\Big(-\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{2}\Big)$ je opravdu pozitivně definitní jádro vyhovující metodě Kernel Ridge Regression.

Kapitola 2

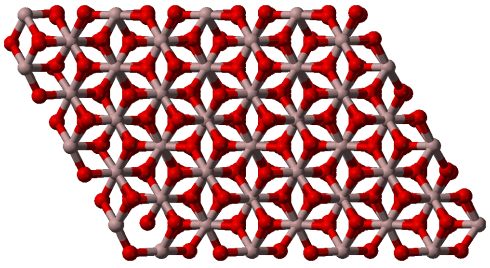
Data

Jako téma pro tuto bakalářskou práci byla vybrána úloha ze soutěže NOMAD 2018 probíhající na serveru Kaggle - viz. [9]. Cílem soutěže bylo predikovat ze vstupních dat pomocí nejrůznějších metod strojového učení dva výstupní parametry (*Bandgap energy* a *Formation energy*) jednotlivých materiálů potencionálně vhodných k výrobě solárních panelů, kterým se říká *světlopropustné vodiče*. V této části se blíže zaměříme na jednotlivá vstupní data (parametry popisující každý jednotlivý materiál - tzv. *deskriptory*), jakož i na právě předpovídané energie.

2.1 Materiály - krystaly

Jak již bylo zmíněno, v této práci bylo pracováno s materiály potencionálně vhodnými k výrobě solárních panelů, kreté se sestavují z fotovoltaických článků. Konkrétně se jednalo o sloučeniny (krystaly) s chemickým vzorcem $(Al_xIn_yGa_z)_{2N}O_{3N}$, kde x, y, z mohou být různá čísla s hodnotou mezi 0 a 1, vždy však splňují podmínku $x + y + z = 1$. O těchto materiálech je známo, že jsou dobré vodiče a zároveň mají velmi nízkou absorbci energie ve viditelném spektru. Vzájemná kombinace těchto vlastností je důležitá nejen pro výrobu již zmíněných fotovoltaických článků, ale také pro výrobu diod emitujících světlo, displeje, tranzistory atd. Nicméně materiálů, které vyhovují oběma požadavkům na dobrou elektrickou vodivost a současně jsou dostatečně světlo-propustné, není mnoho.

V dnešním světě, kdy rostou nároky na množství vyráběné energie, a zároveň je kladen velký důraz na ekologii, se vkládá mnoho úsilí do vývoje co nejvhodnějších materiálů právě pro výrobu fotovoltaických článků, jelikož solární energie patří k potencionálně nejpoužitelnějším zdrojům elektrické energie. A právě sesqui-oxidy hliníku, gallia a india mohou pomoci při této nelehké snaze.



Obrázek 2.1: Krystalická struktura hlavního zástupce, Al_2O_3. Červená - kyslík, šedá - hliník. *Zdroj:* [16].

2.2 Bandgap energy

Bandgap energy - česky *pás zakázaných energií*, nebo zkráceně *zakázaný pás* - je označení pro energetické pásmo v elektronovém obalu atomu, ve kterém se nenachází žádné elektrony. Je to vrstva mezi posledním valenčním elektronem a prvním elektronem, který se může podílet na elektrické vodivosti atomu, tedy elektronem, který může atom vypustit za účelem ionizace a následného pohybu v krystalové mřížce, aby sloužil jako nositel náboje.

Hodnota energetické hladiny bandgap musí být u světlopropustných vodičů větší než energie fotonů dopadajících ze slunečního záření na tyto vodiče. Pokud tomu tak skutečně je, materiál neabsorbuje žádnou energii tohoto světla a fotony mohou projít skrz. V této práci je udávaná v jednotkách eV.

2.3 Formation energy

Formation energy - česky *formační energie* - je energie, která je potřebná ke vzniku konkrétní krystalové konfigurace z jednotlivých volných atomů. Výpočet formační energie si předvedeme na vymyšleném atomu, který má chemický vzorec ABC_3. Označme formační energii této látky H_l. Potom

$$H_l = E(ABC_3) - \mu_A - \mu_B - 3\mu_C, \tag{2.1}$$

kde $E(ABC_3)$ je celková energie sloučeniny ABC_3, μ_A je chemický potenciál látky A, stejně pak pro B a C. Formační energie indikuje stabilitu systému. Proto je také důležitá pro materiály, kterými se zabýváme. Všude v této práci se formační energie vyskytuje v jednotkách eV/atom.

2.4 Trénovací, testovací a validační data

Ke každé metodě využívající strojové učení jsou zapotřebí tzv. *trénovací* a *testovací data*. Na trénovacích datech se program naučí jednotlivé závislosti mezi popisujícími vlastnostmi a výstupy. Tyto mnohdy skryté závislosti jsou poté aplikovány na testovací data, ke kterým se vyhodnotí příslušné výsledky. V této práci bylo pracováno s 2400 materiály (různě se lišícími krystaly), jakožto trénovacími daty, a 600 materiály zkušebními, u kterých bylo nutno 2 parametry dopočíst. O těchto dvou parametrech byla již řeč v částech 2.2 a 2.3.

Pro vizualizaci vlastních výsledků budeme ještě používat tzv. *validační data*. To budou data, která budou sloužit k výběru nejvhodnějších parametrů (ve smyslu závislosti) příslušné metody strojového učení. Z trénovacích dat vždy nějakým způsobem vybereme určitou část, kterou určíme jako validační, na zbytku trénovacích dat provedeme výpočet, předpovíme parametry pro validační data a obdržené výsledky porovnáme s přesnými hodnotami, které pro validační data již máme (validační data pocházejí z trénovacích, tudíž máme dostupné přesné hodnoty předpovídaných parametrů).

2.5 Deskriptory

Jak už bylo zmíněno, jako deskriptor bývá označována vlastnost (popř. parametr či stav) materiálu, která slouží k bližší specifikaci a identifikaci. Prvotní počet deskriptorů byl 11. Těmito deskriptory byly: *prostorová skupina, celkový počet atomů, procenta zastoupení prvků hliníku, galia a india, 3 mřížkové vektory a 3 mřížkové úhly*.

Prostorová skupina udává, jakým způsobem je krystalová mřížka symetrická a nabývá hodnot od 1 do 230. Mřížkové vektory, resp. jejich délky, jsou uvedené v jednotkách *angstrém* [Å], přičemž 1Å = 10^{-10} m. Mřížkové úhly jsou pak udávány ve stupních od 0 do 360.

Ke každému materiálu však byla přiložena tabulka s dalšími informacemi, tedy prostorové souřadnice jednotlivých konkrétních atomů v krystalové mřížce každého materiálu. Z těchto dodatečných informací byly posléze vytvořeny další deskriptory. Obecně totiž platí, že počet deskriptorů se může zvyšovat či snižovat. Deskriptory lze mezi sebou různě upravovat či kombinovat.

Na začátku byla snaha vytvářet deskriptory pouze z oněch prvotních 11 údajů. Šlo například o vytvoření jednoho deskriptoru jako rozdíl mezi dvěma prostorovými úhly, jelikož se téměř vždy lišily pouze v řádu desetin či jednotek stupňů. Dalším deskriptorem byl posléze objem jedné prostorové buňky, následně však „znormovaný" počtem atomů konkrétní buňky, neboť z důvodu rozdílných symetrií nebyl počet atomů v jednotlivých konfiguracích připadajících na jednu buňku vždy stejný.

Čím více práce pokračovala, bylo více a více zřejmé, že pouze s deskriptory ze základní tabulky nebude možné dosáhnout výraznějšího zlepšení. Bylo tedy nutné sáhnout k dodatečným souborům, které obsahovaly ke každému materiálu prostorové souřadnice všech atomů v jedné buňce krystalové mřížky. Jedním z prvních pokusů bylo použít jako deskriptor minimum vzdálenosti vazby *kov-kyslík*. Postupným zlepšováním se dospělo až k tomu, že nejlepšími deskriptory byly tzv. *monogramy*, tedy procenta zastoupení jednotlivých chemických látek, navíc však byly také rozlišeny podle vaznosti. Kovy měly zastoupení jako 3, 4 a 5-ti vazné, kyslík byl pak 2, 3, 4 a 5-ti vazný. O tomto postupném zdokonalování programu bude řeč v následují kapitole.

2.6 Ukázka dat

V následující tabulce 2.1 je možno vidět příklady popisu čtyř materiálů. První dva jsou materiály z trénovací sady a další dva jsou pak ze sady testovací. Bližší informace o jednotlivých deskriptorech z této tabulky jsou podány v části 2.5.

Tabulka 2.1: Ukázka dat

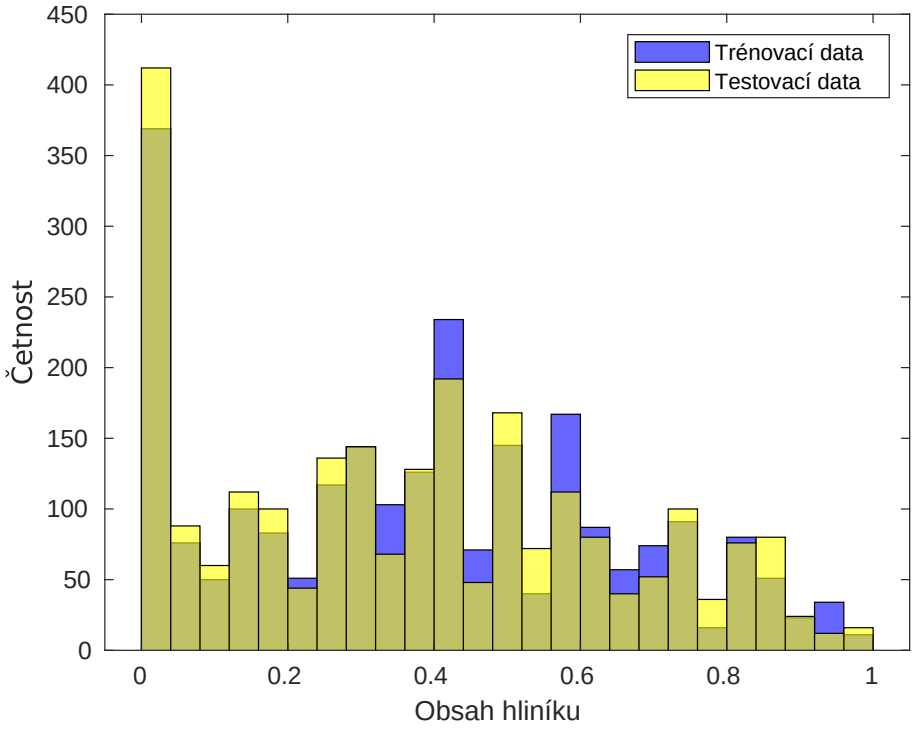
ID	Prostorová skupina	Počet atomů/buňku	*Al* [%]	*Ga* [%]	*In* [%]	Vektor 1	Vektor 2
51	167	30	0,4167	0,3333	0,25	5,0862	5,0858
1236	194	10	0,5	0,0	0,5	3,2964	3,2962
10	206	80	0,75	0,0	0,25	9,3111	9,3105
99	33	40	0,0	0,9375	0,0625	5,1569	8,8529

Vektor 3	Úhel 1 [°]	Úhel 2 [°]	Úhel 3 [°]	Formační energie	Zakázaný pás
13,6981	89,9942	90,0065	120,001	0,1608	2,5143
12,1828	90,0168	90,0124	119,9929	0,6121	1,42
9,3108	90,0016	90,0026	89,9993	—	—
9,4691	89,9966	90,0015	90,0017	—	—

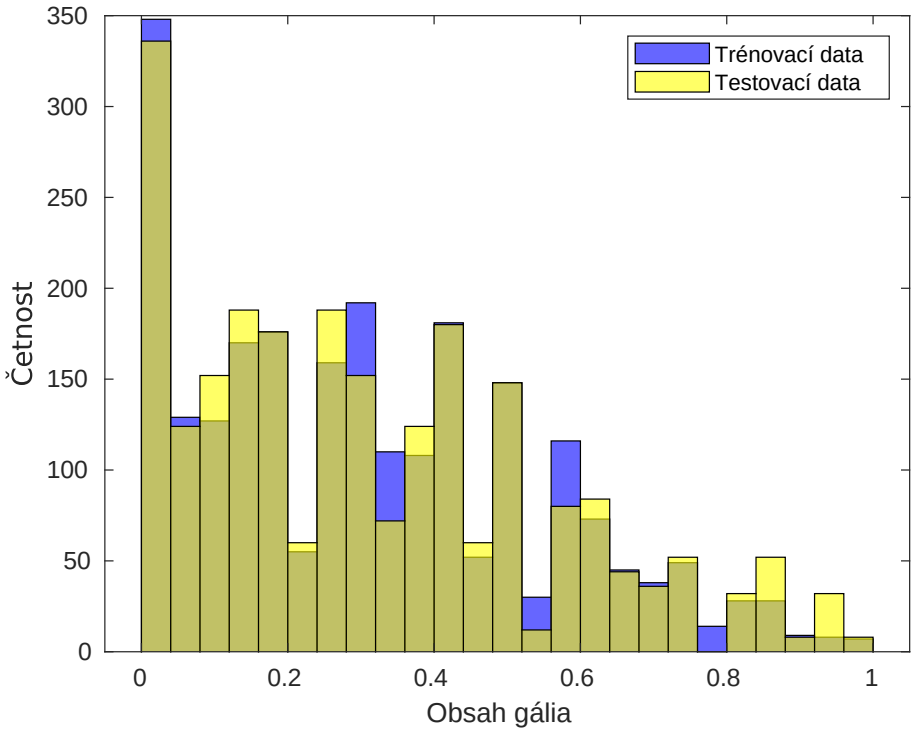
2.7 Vizualizace dat

V této části se pokusíme o jednoduchou vizualizaci dat, se kterými pracujeme. Zaměříme se jak na trénovací, tak na testovací data.

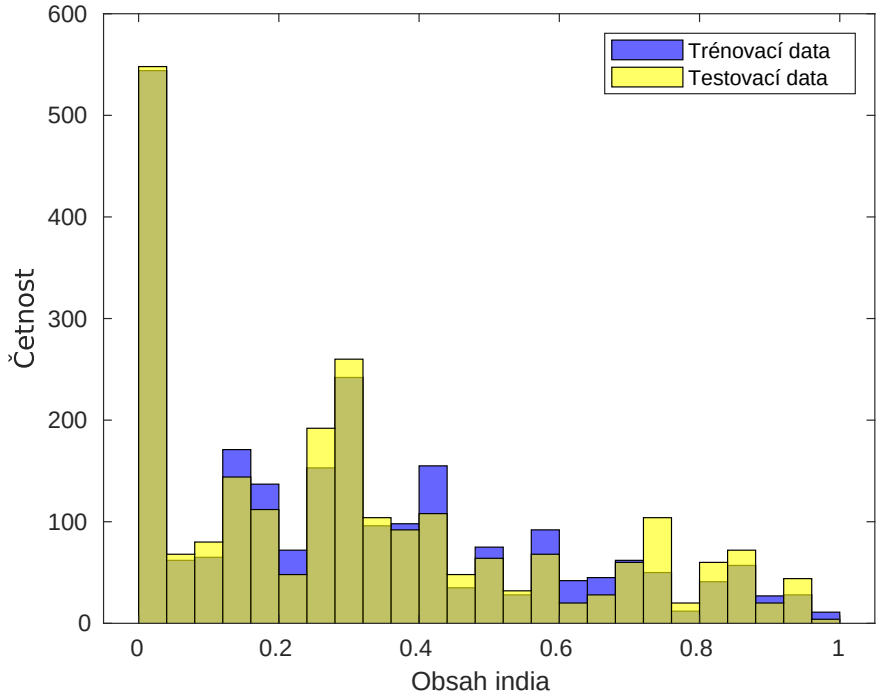
Na prvních třech histogramech můžeme vidět srovnání trénovacích dat a k nim příslušně přeškálované hodnoty pro data testovací (pro trénovací jsou údaje přesné, pro testovací jsou četnosti pro histogram vynásobené 6x - trénovacích dat je šest krát více než testovacích).



Obrázek 2.2: Histogram materiálů podle obsahu hliníku

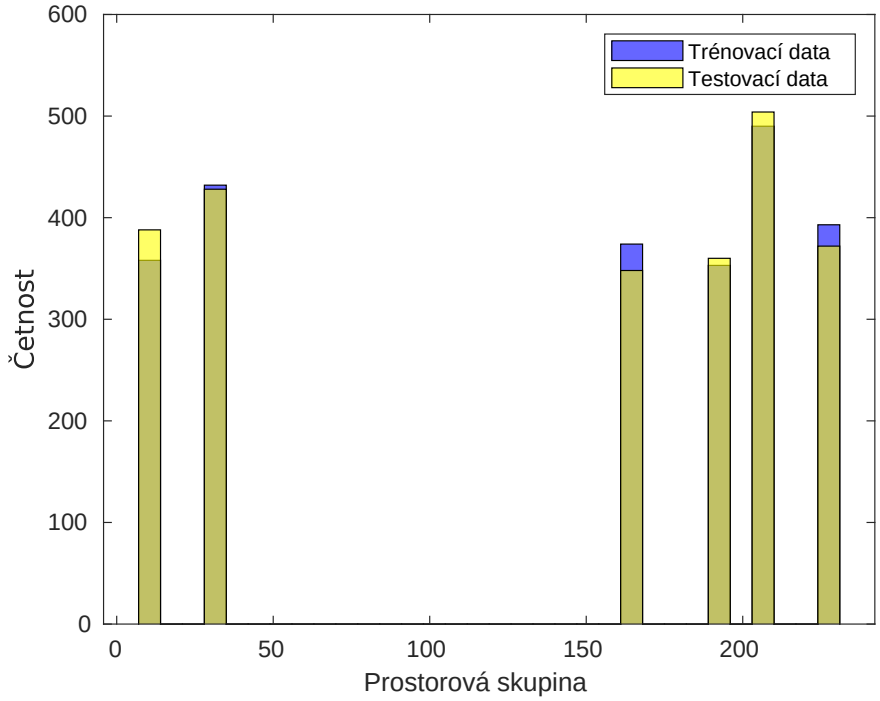


Obrázek 2.3: Histogram materiálů podle obsahu gália



Obrázek 2.4: Histogram materiálů podle obsahu india

Na posledním histogramu 2.5 je možné vidět, že jsou zastoupené jen některé prostorové skupiny. Konkrétně se jedná o skupiny 12, 33, 167, 194, 206 a 227. Testovací data jsou opět přeškálovaná.



Obrázek 2.5: Histogram materiálů podle prostorové skupiny

Kapitola 3

Praktická část

Cílem této bakalářské práce bylo osvojit si některou z metod strojového učení na teoretické úrovni a tu posléze aplikovat na reálný problém. Vybrána byla metoda Kernel Ridge Regression. A právě v této kapitole budou prezentovány výsledky získané vytvořením vlastního programu. Tento program byl postupem času zdokonalován a vylepšován. Kapitola 3 se bude skládat z popisu samotného programu, následně bude vysvětlen způsob vyhodnocování kvality obdržených výsledků a hlavní částí se stane několik pokusů, které budou detailně popsány.

3.1 Program

Pro účely této bakalářské práce byl vytvořen program v programovacím jazyce Matlab. Samotná výpočetní metoda byla napsaná jako samostatná funkce, která byla vždy implementována do příslušného skriptu. Dále v této části je možno vidět kód samotné funkce. Kód funkce byl převzat z [6], jelikož psát znovu již napsaný kód by nepřineslo žádný přínos. Nicméně tento kód jen krátce okomentujeme..

Na řádcích 3-10 je prováděna kontrola, zda jsou všechna vstupní data stejných rozměrů a zda tedy bude možné provést výpočet. Na řádcích 16 až 21 pak probíhá výpočet matice K dle značení užitého v (1.20). V našem kódu má tedy matice K označení dále *x_in*. Jak již bylo zmíněno v části 1.5, k výpočtu využíváme Gaussovo jádro, což lze vidět konkrétně na řádku 19. Na řádcích 22 až 25 pak probíhá pouze kontrola symetričnosti matice K.

Samotný výpočet výsledků probíhá od řádku 28. Po kontrole (probíhající na řádcích 29 až 32), zda bude smět být výpočet proveden, se vypočítal koeficient Σ dle značení z rovnice (1.22) (v naší funkci vystupuje jako „alpha" na řádku 33). V cyklu na řádcích 34 až 41 se uskutečňuje poslední fáze celého výpočtu, tedy výpočet násobení mezi κ z (1.23) a již zmíněným Σ. Konečný výsledek je pak výstupem ve formě řádkového vektoru dat.

```
1 function final_ans = KernelRidge(in_data,out_data,test_data,lamda)
2
3 if size(in_data,2) ~= size(out_data,2)
4     fprintf('\nTotal number of points for function input and output
        are unequal');
5     fprintf('\nExitting program');
6     return
7 elseif size(test_data,1) ~= size(in_data,1)
8     fprintf('\nTest data and Input data are of unequal dimensions');
9     fprintf('\nExitting program')
```

```
10      return
11  else
12      tot_data = in_data;
13      vec_test = test_data;
14      x_in = zeros(size(tot_data,2),size(tot_data,2));
15      %% x_in(i,j) = x_in(j,i) -- Using symmetry of the Kernel
16      for row = 1:size(x_in,2)
17          for col = 1:row
18              temp = sum((tot_data(:,row)-tot_data(:,col)).^2);
19              x_in(row,col) = exp(-temp/2);
20          end
21      end
22      x_in = x_in + x_in';
23      for count = 1:size(x_in,2)
24          x_in(count,count) = x_in(count,count)/2;
25      end
26
27      %% Calculating alpha and the final answer
28      final_ans = zeros(1,size(vec_test,2));
29      if det(x_in + lamda*eye(size(x_in))) > 1e10
30          fprintf('\nThe kernel matrix is poorly scaled. Choose better
                parametr.');
31          return
32      end
33      alpha = inv(x_in + lamda*eye(size(x_in)))*out_data';
34      for count1 = 1:size(vec_test,2)
35          temp = 0;
36          for count2 = 1:size(alpha,1)
37              temp1 = sum((vec_test(:,count1)-tot_data(:,count2)).^2);
38              temp = temp + alpha(count2)*exp(-temp1/2);
39          end
40          final_ans(count1) = temp;
41      end
42  end
```

3.2 Výpočet chyby

Pro porovnání výsledků získaných nejrůznějšími metodami strojového učení se skutečnými hodnotami se využívá mnoho druhů výpočtů odchylek, tedy chyb. V soutěži na serveru Kaggle byla používána tzv. *RMSLE (Root Mean Squared Logarithmic Error)* s výpočtem

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\ln(y_i+1)-\ln(a_i+1)\right)^2}, \tag{3.1}$$

kde

- n je počet materiálů

- $\ln(x)$ je přirozený logaritmus čísla x
- y_i je předpovězená hotnota a a_i je skutečná hotnota paramteru i-tého materiálu.

Naproti tomu chyba *RMSE (Root Mean Squared Error)* se počítá podle vztahu

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - a_i\right)^2} \tag{3.2}$$

při zachovaném značení z rovnice (3.1).

Nutno podotknout, že jelikož výpočet dvou energií byl prováděn odděleně (nezávisle), i obě chyby se musely vypočítat pro každý ze dvou parametrů zvlášť a celková výsledná chyba byla následně jejich průměrem, tedy

$$\mathrm{RMSLE}_{\text{výsledná}} = \frac{\mathrm{RMSLE}_{\text{bandgap}} + \mathrm{RMSLE}_{\text{formation}}}{2}.$$

Zcela totožně by se vypočítala celková výsledná chyba pro RMSE.

Chyba RMSLE byla v soutěži použita z následujícího důvodu. Cílem soutěže bylo předpovědět dva parametry, které jsou na sobě nezávislé. Bylo zjištěno, že velikost chyby u předpovědí formační energie a zakázaného pásu se v průměru liší o jeden řád (jelikož i hodnoty energií jsou přibližně o jeden řád rozdílné). Kdybychom tedy používali chybu RMSE, docházelo by ke většímu zohlednění dosažených výsledků pro hodnoty zakázaného pásu, jelikož právě ony jsou o řád vyšší než hotnoty formační energie. Výrazné zlepšení právě v předpovědi pro formační energii by ve výsledku nehrálo téměř žádnou roli, pokud by se nezlepšily výsledky pro druhý parametr. Naopak použijeme-li pro výpočet chybu RMSLE, zlogaritmování ve vztahu (3.1) „odstraní" řádový rozdíl a dílčí chyby se stanou srovnatelně velké. Celková výsledná chyba pak bude reflektovat ať už zlepšení či zhoršení v jedné nebo druhé dílčí chybě velmi podobně.

Zaveďme ještě značení pro chyby v této části práce. Dosažené výsledky budeme prezentovat za pomoci chyby RMSLE, a to jak celkových, tak také dílčích chyb. K tomu budeme využívat následující značení (znovu nutno podotknout, že následující chyby jsou vždy RMSLE, tedy počítají se dle vztahu (3.1)):

- **ERR-bg** bude označení pro chybu *bandgap energy*, tedy zakázaného pásu
- **ERR-fe** bude označení pro chybu *formační energie*
- **ERR** bude označení pro celkovou chybu, která se vypočítá jako $\mathrm{ERR} = \frac{\text{ERR-bg} + \text{ERR-fe}}{2}$.

3.3 Standardizace dat

Ve statistických modelech je důležitou součástí postup tzv. *standardizace dat*. Cílem standardizace je „odstranit" rozdílnosti v „měřítku" mezi jednotlivými daty. Ne vždy je však vhodná, někdy může ztratit informaci, kterou data mohou nést. Typů standardizací je mnoho, např. standardizace směrodatnou odchylkou, min-max standardizace nebo standardizace na součet v řádku. V této práci byla použita patrně nejpoužívanější standardizace, tedy směrodatnou odchylkou. V každém sloupci (v jednom parametru u všech materiálů) byla od dat odečtena společná střední hodnota a následně byla data podělena odmocninou z rozptylu. Nová data měla ve výsledku nulovou střední hodnotu a rozptyl dat byl roven jedné.

Postup standardizace si názorně ukážeme na příkladu. Mějme tedy 5 materiálů a k nim tři náhodné parametry.

Tabulka 3.1: Ukázka dat před standardizací

ID	Parametr 1	Parametr 2	Parametr 3
1	2,1530	15,5893	0,9121
2	4,8078	14,6769	1,4835
3	3,8121	13,2846	3,5637
4	0,0367	8,2716	3,4255
5	3,4002	16,5855	1,6097

Nyní na těchto datech provedeme standardizaci popsanou výše. Níže můžeme vidět kód k této standardizaci.

```
for j = 1:3
    x_mean(j) = mean(x(:,j));
    x_working(:,j) = x(:,j)-x_mean;
    x_var(j) = norm(x_working(:,j),2);
    x_working(:,j) = x_working(:,j)/x_var(j);
end
```

A výsledná standardizovaná data s odpovídajícími vlastnostmi - tedy s nulovou střední hodnotou a jednotkovým rozptylem v každém sloupci - vidíme v tabulce 3.2.

Tabulka 3.2: Ukázka dat po standardizaci

ID	Parametr 1	Parametr 2	Parametr 3
1	-0,1878	0,2927	-0,5306
2	0,5358	0,1527	-0,2950
3	0,2644	-0,0609	0,5627
4	-0,7646	-0,8300	0,5058
5	0,1522	0,4455	-0,2429

3.4 Vysvětlení k pokusům

Dříve, než přistoupíme k vysvětlování a popisu jednotlivých pokusů, je potřeba zmínit několik obecných věcí, které se vztahují ke všem z nich.

V první řadě je to srovnání se soutěží NOMAD 2018. V části 3.2 bylo zavedeno značení pro jednotlivé chyby a také bylo ukázáno, jak se jednotlivé chyby vypočítávají. Pokud bychom chtěli srovnávat výsledky v rámci soutěže, pak bychom vždy používali jako trénovací data oněch 2400 materiálů, které sloužily právě pro účel trénování. Testování metody bychom poté prováděli na 600 testovacích datech. Tento způsob realizace testování by bylo možné provádět také v této práci. Nicméně to možné není, a to z následujícího důvodu. K testovacím datům ze soutěže nejsou známy skutečné výsledky. Nebylo by tedy možné porovnávat obdržená data s přesnými hodnotami. Jediný způsob, jak prezentovat kvalitu obdržených výslekdů, je tedy vybrat ze známých 2400 trénovacích materiálů určitou část a vytvořit si „vlastní soutěž".

Zde nastává další problém, a to jakým způsobem vybrat ze známých materiálů data tak, aby testovací soubor tvořil reprezentativní vzorek. Mezi materiály je totiž zastoupeno množství takových sloučenin,

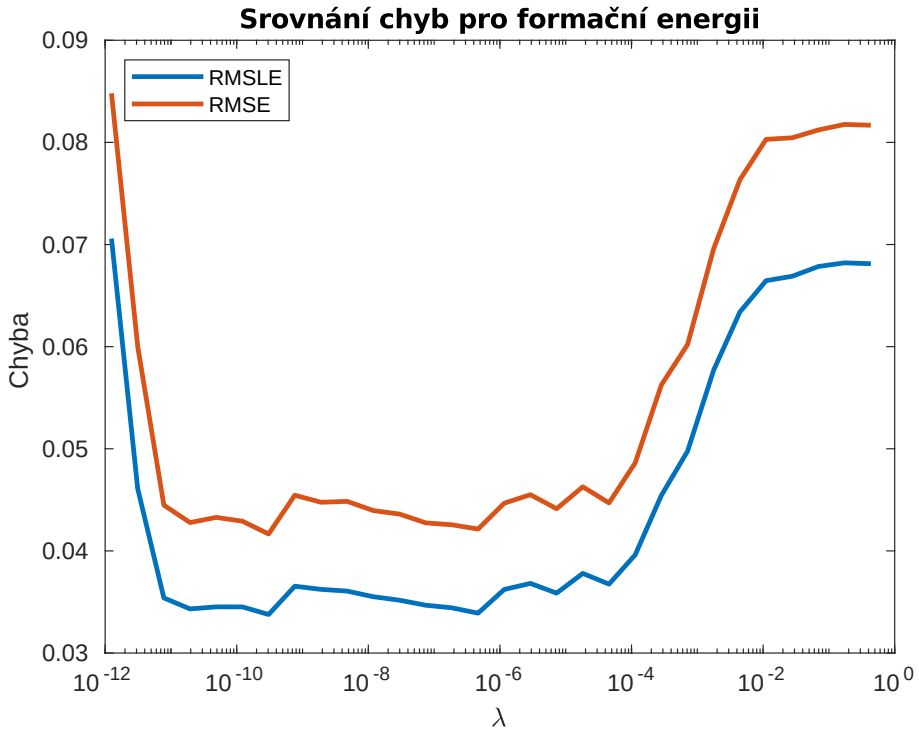
u kterých je předpověď obtížnější než u ostatních, a chyba předpovědi bývá proto zpravidla větší. K problému proto přistoupíme následovně. Z 2400 materiálů, pro které známe přesné hodnoty dvou výstupních parametrů, vybereme vždy náhodně 200 vzorků, které budeme považovat za testovací. Budeme mít tedy 2200 materiálů trénovacích a oněch 200 materiálů testovacích, pro které budeme předpověď provádět. Vyhodnocením poté bude chyba vypočítaná dle známého postupu vysvětleného v části 3.2.

Aby však nedocházelo k přílišné závislosti na náhodném výběru, bude náhodný výběr proveden několikrát a následně bude vytvořen průměr chyb jednotlivých výběrů. Výsledná chyba pak bude právě tímto průměrem.

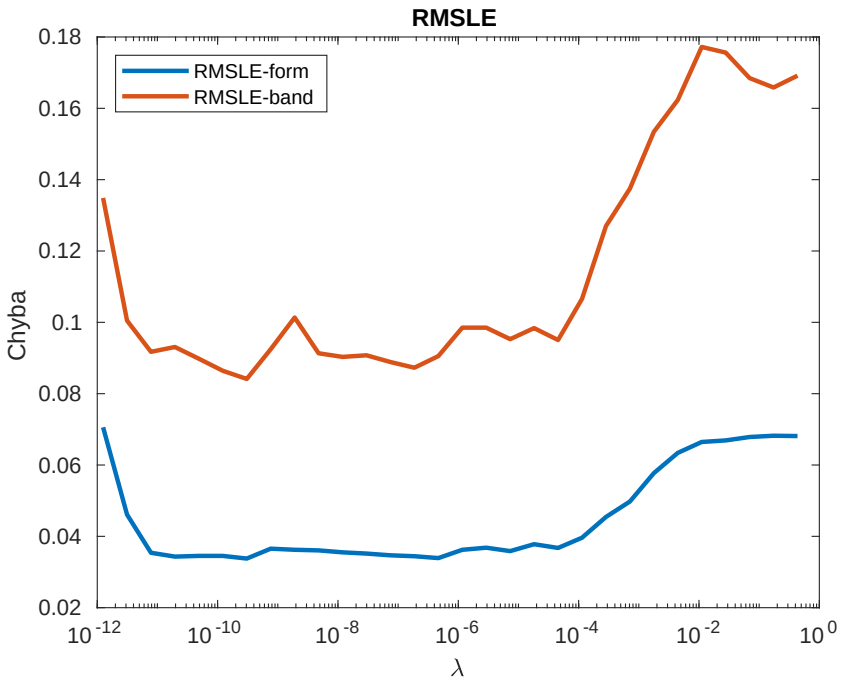
3.5 Pokus č.1

Prvním pokusem o předpověď nových hodnot pro materiály bylo prosté spuštění funkce Kernel Ridge Regression popsané v části 3.1 na původní data bez další úpravy. Proběhlo tedy načtení dat a provedení náhodného výběru testovacích vzorků (popsané v části 3.4). Následovala standardizace (dle 3.3) a pak již proběhl výpočet.

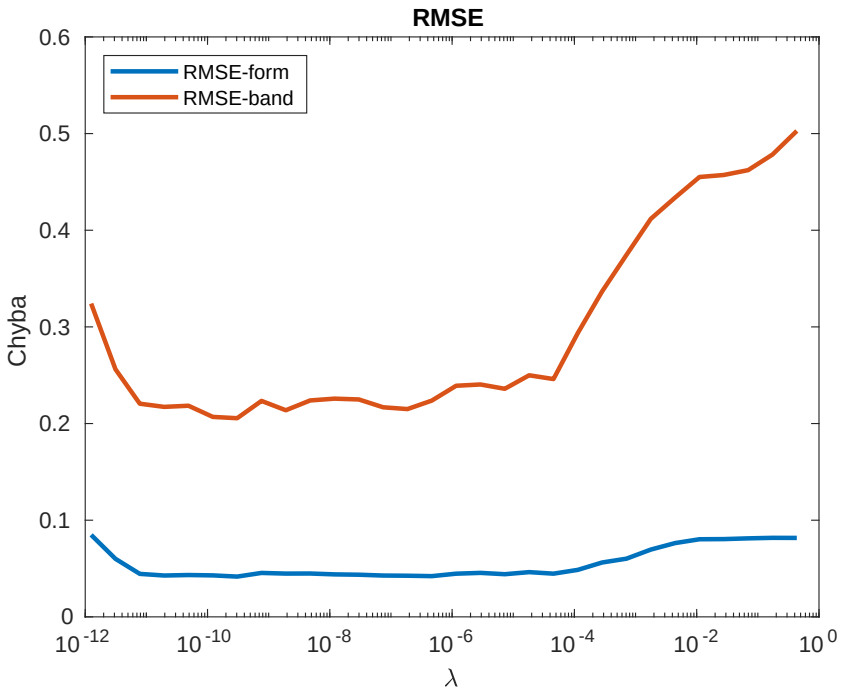
Na datech, ze kterých jsou vykreslené následující grafy, byl prováděn náhodný výběr dvacetkrát.



Obrázek 3.1: Srovnání chyby RMSLE a RMSE



Obrázek 3.2: Chyba RMSLE pro různou volbu parametru λ



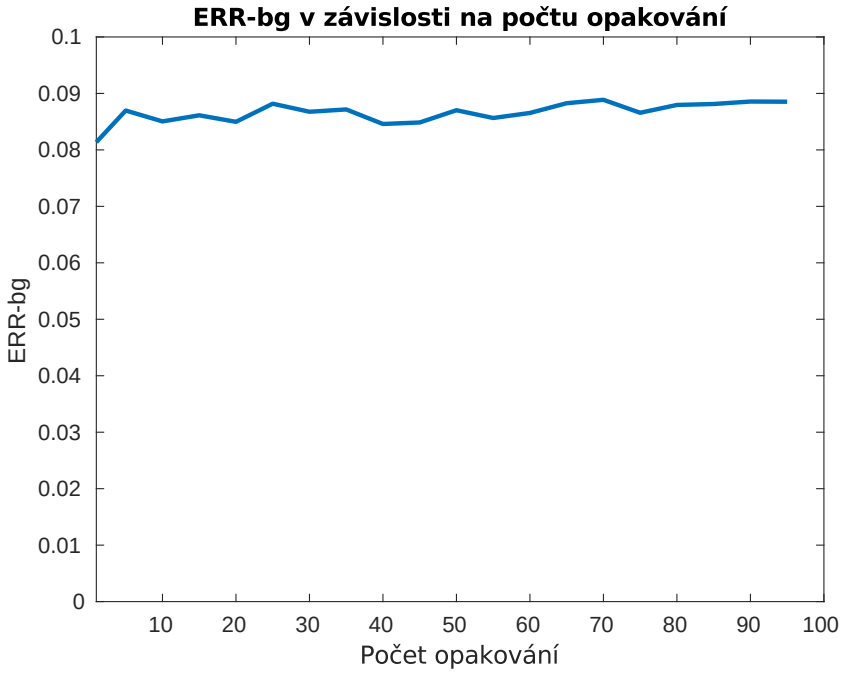
Obrázek 3.3: Chyba RMSE pro různou volbu parametru λ

Pro první pokus vyšla chyba ERR-bg = 0,0841, ERR-fe = 0,0338 a celková chyba ERR = 0,0590. V soutěži NOMAD by to odpovídalo přibližně 564. místu z 882 účastníků.

Tabulka 3.3: Výsledky prvního pokusu

Pokus	ERR-fe	ERR-bg	ERR	umístění (z 882)
1	0,0338	0,0841	0,0590	**614.**

Bylo by zároveň vhodné vysvětlit, proč nám postačuje provádět náhodný výběr dvacetkrát. Na následujícím obrázku 3.4 můžeme vidět, jak se mění chyba (konkrétně ERR-bg) v závislosti na zvětšujícím se počtu opakování náhodného výběru. Z obdržených výsledků vyplývá, že zvyšující se počet opakování náhodného výběru nevede k výraznému zlepšování výsledné chyby. Proto v rámci zrychlení výpočtu bylo rozhodnuto o používání počtu 20 náhodných výběrů.



Obrázek 3.4: Závislost ERR-bg na počtu opakování náhodného výběru pro $\lambda = 1,25 \cdot 10^{-8}$

3.6 Pokus č.2

V rámci druhého pokusu bylo cílem vyzkoušet metodu na lehce upravený datech. Pozorováním bylo například zjištěno, že ve většině případů se první dva úhly v krystalové mřížce rovnají přibližně devadesáti stupňům. Prvním nápadem na modifikaci deskriptorů bylo vytvořit nový deskriptor, a to absolutní hodnotu rozdílu prvních dvou úhlů. Druhým deskriptorem se stal „normovaný objem buňky" - tedy objem jedné krystalové buňky konkrétního materiálu podělený příslušným počtem atomů v této buňce.

Pro výpočet chyb ERR-bg, ERR-fe a ERR byly tedy použity původní deskriptory spolu s dvěma výše zmíněnými. Byl proveden výpočet těchto deskriptorů, standardizace dat dle vysvětlení v části 3.3, výběr nejlepších parametrů λ a samotný výpočet výstupních parametrů.

Hodnota nejideálnějšího parametru λ pro formační energii vyšla $1 \cdot 10^{-10}$, pro zakázaný pás pak $4,1 \cdot 10^{-7}$. Hodnoty chyb jsou k vidění v následující tabulce 3.4, která srovnává výsledky prvního a druhého pokusu.

Tabulka 3.4: Výsledky prvního a druhého pokusu

Pokus	ERR-fe	ERR-bg	ERR	umístění (z 882)
1	0,0338	0,0841	0,0590	**614.**
2	0,0321	0,0796	0,0559	**478.**

Lze vidět, že oproti prvnímu pokusu došlo ke zlepšení. Nicméně toto zlepšení není nikterak výrazné. Bylo tedy zřejmé, že pro zpřesnění předpovědí bude nutné vytvořit zcela nové deskriptory, které budou vycházet z dodatečných informací o prostorových polohách jednotlivých atomů v krystalické mřížce každého materiálu.

3.7 Pokus č.3

Jak již bylo zmíněno v části 3.6, pro vytvoření přesnější předpovědi parametrů můsí být využity informace v dodatečných souborech, které obsahují prostorové souřadnice jednotlivých atomů v krystalické mřížce. Po lehkém nastudování základních chemických principů bylo rozhodnuto, že jako nové deskriptory vyzkoušíme minima vzdáleností atomů kovů od kyslíků i kovů mezi sebou. Hledáme tedy minimální délku vazby $X - O$, kde X je atom hliníku, gália či india, či vazby $X - Y$, kde opět za X a Y bereme všechny kombinace kovů. Nezapomínáme také na vazbu $O - O$. Celkem tedy dostáváme 10 nových deskriptorů.

Při hledání minimálních vzdáleností nesmíme zapomínat na periodičnost krystalů! V souborech se souřadnicemi jednotlivých atomů jsou vždy uvedeny pouze souřadnice jedné buňky. Při počítání vzdáleností však musíme nějakým způsobem zohlednit onu periodičnost. V našem případě jsme použili následující přístup. Ke každé buňce jsme do všech směrů vedle ní „nakopírovali" za pomoci mřížkových vektorů další atomy. Vytvořili jsme si tedy několik buněk krystalu u sebe (konkrétně 27 - okolo prostřední buňky vždy $3 \times 3 \times 3$ buňky). Vzdálenosti jsme pak počítali pro všechny atomy prostřední buňky od všech ostatních atomů.

Máme tedy 13 deskriptorů původních (z části 3.6) a k nim přidáváme 10 deskriptorů nových. Pro tento pokus jich tedy celkem máme 23.

Provedeme opět již několikrát komentovaný výpočet (standardizace, výběr nejlepšího $\lambda, \dots$). Hodnota nejvhodnějšího parametru λ pro formační energii vyšla $2,6 \cdot 10^{-5}$, pro zakázaný pás pak $4,1 \cdot 10^{-7}$. Obdržené výsledky jsou k porovnání se všemi předchozími výsledky v následující tabulce.

Tabulka 3.5: Výsledky prvního, druhého a třetího pokusu

Pokus	ERR-fe	ERR-bg	ERR	umístění (z 882)
1	0,0338	0,0841	0,0590	**614.**
2	0,0321	0,0796	0,0559	**478.**
3	0,0311	0,0822	0,0567	**507.**

V tabulce 3.5 můžeme vidět, že oproti druhému pokusu došlo ke zlepšení pouze v předpovědi formační energie, zatímco u zakázaného pásu došlo ke zhoršení. Tušíme tedy, že práce s dodatečnými soubory povede ke zlepšení, nicméně musíme dojít k vhodnějším deskriptorům.

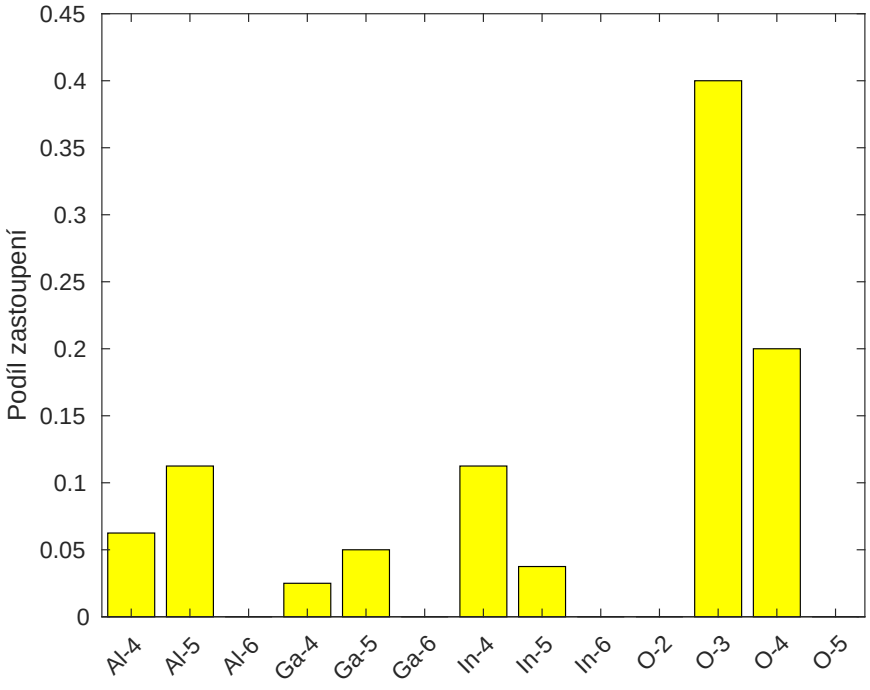
3.8 Pokus č.4

V této části práce narazil autor této práce na článek [12]. V něm jsou popsané postupy trojice, která se umístila v samotné soutěži NOMAD 2018 na prvních třech místech. Vítěz soutěže rovněž používal metodu Kernel Ridge Regression. K dosažení svých výsledků používal takzvané monogramy, případně bigramy. Zkusíme se nejprve podívat na postup vítěze soutěže a jeho výsledky replikovat. K tomu se budeme muset seznámit s monogramy, tedy s tím, co jsou, jak se získají a proč mohou vést ke zlepšení výsledků předpovědi.

3.8.1 Monogramy

V krystalické látce jsou mezi různými atomy různé vazby. Zároveň i atomy mohou mít různé počty vazeb se svými sousedy. Dokonce i atomy jedné látky se mohou vyskytovat ve více variantách atomů co se týče vaznosti. A právě vaznost jednotlivých kovů má vliv například na vodivost. Souvisí však také s námi předpovídanými energiemi.

Zkoumáním struktury některých krystalických mřížek bylo zjištěno, že vaznost tří kovů vyskytujících se ve sloučeninách, se kterými pracuje soutěž NOMAD 2018, tedy s hliníkem, gáliem a indiem, se pohybuje různě od čtyř do šesti. Vaznost kyslíku je pak v rozmezí od dvou do pěti. Když si tedy uděláme tabulku či histogram s hodnotami procentuálního zastoupení příslušných atomů podle počtu vazeb v dané látce, dostaneme tzv. *monogram*. Celkem tedy dostáváme 13 nových deskriptorů pro každou sloučeninu. Na následujím histogramu 3.5 je možné vidět vaznost jednotlivých atomů pro sloučeninu číslo 15. Zkratka udává prvek a číslo za pomlčkou příslušnou vaznost.



Obrázek 3.5: Monogram pro sloučeninu č.15.

3.8.2 Výroba monogramů

Již víme, co jsou to monogramy. Nicméně mnohem náročnější část nastává tehdy, když chceme tyto monogramy sestavit. Určit vaznost jednotlivých atomů pouze ze znalosti prostorových souřadic atomů jedné krystalické buňky není triviální věc. Řešení tohoto problému existuje, dokonce postupů může být více. My zde popíšeme ten, který byl použit v této práci.

V první řadě se použije postup kopírovaní souřadnic atomů do okolí jedné krystalové buňky popsaný v části 3.7. Poté probíhal výpočet následujím způsobem. Pro každý atom prostřední buňky se našla minimální vzdálenost od všech ostatních atomů. Jakmile jsme zjistili minimum, spočítali jsme, kolik dalších atomů je blíž než 1,3 násobek této minimální vzdálenosti. Takto jsme určili, s kolika ostatními atomy tento atom pravděpodobně sdílí vazbu. Tento postup nemusí být vždy účinný, jelikož hodnota 1,3 násobku byla zvolena intuitivně, nicméně vede k očividnému zlepšení ve finální předpovědi. Takto zvolený výpočet je také rychlý a není nikterak složitý. Výpočet byl proveden pro každou sloučeninu a výsledky byly uloženy do zvláštního souboru, který se pak v pozdějším výpočtu pouze načetl.

3.8.3 Provedení výpočtu

Když máme nyní vypočtené hodnoty monogramů pro všechny sloučeniny, lze přistoupit k samotnému výpočtu předpovídaných energií. Jako deskriptory byly použity původní informace, přidán byl pouze objem buňky komentovaný v části 3.6 a výše zmíněné hodnoty monogramů. Dohromady tedy 25 deskriptorů. Výpočet byl proveden obvyklým způsobem (standardizace, výběr nejlepšího $\lambda, \dots$). Výsledky jsou k nahlédnutí v následující tabulce 3.6.

Tabulka 3.6: Výsledky pro čtvrtý pokus

Energie	**Příslušné λ**	**Chyba**
Formační energie	$3{,}11 \cdot 10^{-7}$	0,0202
Zakázaný pás	$2{,}71 \cdot 10^{-7}$	0,0711

Celková chyba tedy vyšla 0,0457. Srovnání s předchozími pokusy je možné nahlédnout v tabulce 3.7.

Tabulka 3.7: Výsledky prvního až čtvrtého pokusu

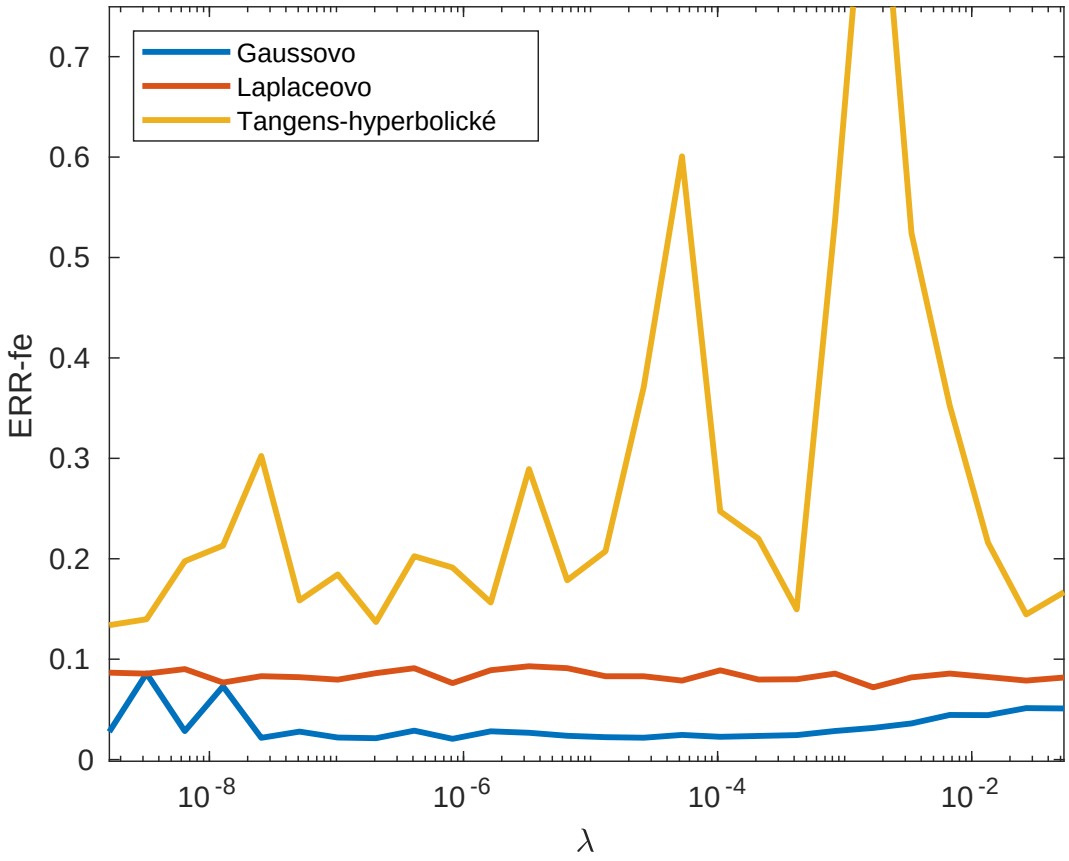
Pokus	**ERR-fe**	**ERR-bg**	**ERR**	**umístění (z 882)**
1	0,0338	0,0841	0,0590	**614.**
2	0,0321	0,0796	0,0559	**478.**
3	0,0311	0,0822	0,0567	**507.**
4	0,0202	0,0711	0,0457	**25.**

Jak můžeme vidět, ke zlepšení opravdu došlo. V celkovém umístění to bylo zlepšení velmi velké, přestože jednotlivé chyby byly zlepšeny v řádech setin. Také si můžeme uvědomit, že v soutěži se dostat k hodnotám 3. pokusu nemuselo být těžké. Dosáhnout však ztlačení chyb pod hodnoty 0,03 pro formační energii a 0,08 pro zakázaný pás dokázalo jen málo účastníků. Z toho je možno usoudit, že obě dvě energie, které chceme předpovídat, velmi souvisí s vazností atomů a jejich vazbami obecně.

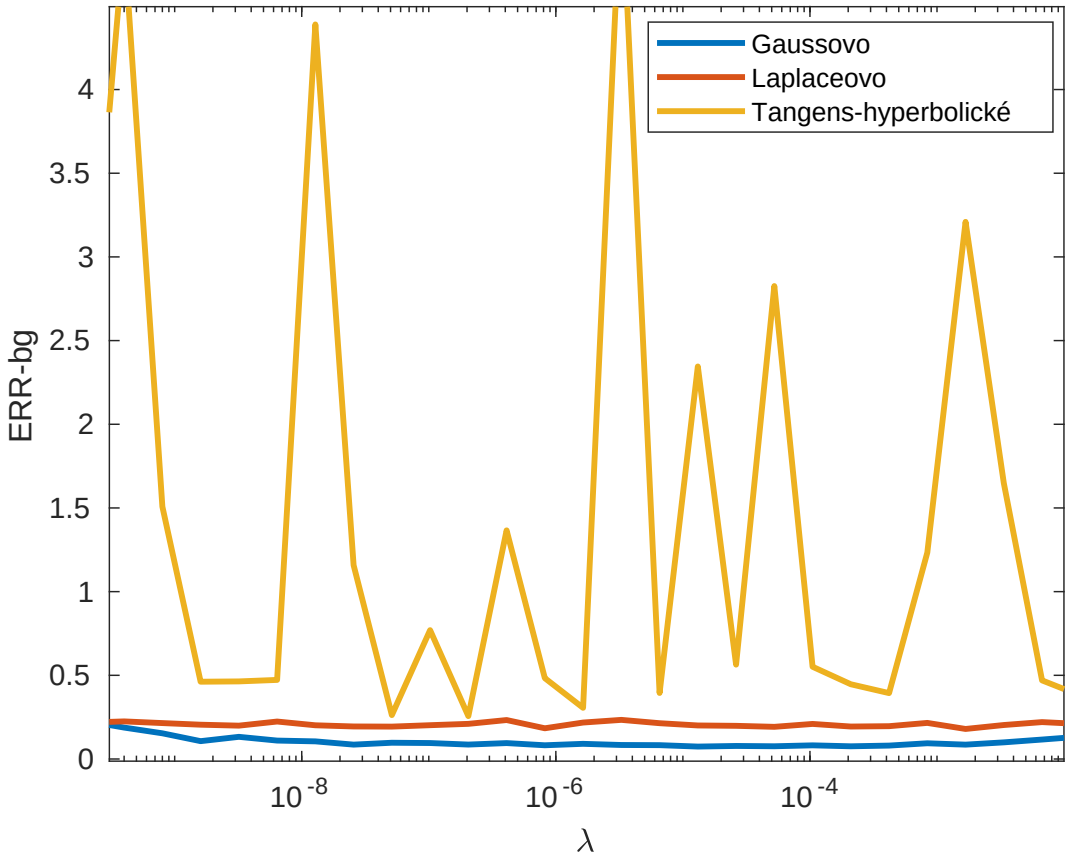
3.9 Změna jádra v metodě Kernel Ridge Regression

Jak již bylo zmíněno v části 1.5, v praktických výpočtech bylo používano jádro Gaussovo. Zkusme však srovnat výpočty i s jinými jádry. Vezměme další dvě jádra, a to Laplaceovo a tangens-hyperbolické jádro a zkusme srovnat výsledky jednotlivých postupů.

Výpočet provedeme na pokusu ze sekce 3.8. Provádíme tedy ten samý výpočet, nicméně zaměňujeme postupně jádro Gaussovo za Laplaceovo a tangens-hyperbolické. Obdržené výsledky je možné vidět na následujících grafech 3.6 a 3.7.



Obrázek 3.6: Chyba ERR-fe v závislosti na λ pro různá jádra



Obrázek 3.7: Chyba ERR-bg v závislosti na λ pro různá jádra

Číselné hodnoty chyb, parametrů λ a konkrétní předpisy jader je možné nahlédnout v tabulce 3.8.

Tabulka 3.8: Výsledky pro srovnání různých jader

Předpis jádra	λ **pro ERR-fe**	**ERR-fe**	λ **pro ERR-fe**	**ERR-bg**	**ERR**
$k(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2}\right)$	$8{,}19 \cdot 10^{-7}$	0,0207	$1{,}3107 \cdot 10^{-5}$	0,0742	0,0474
$k(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|}{2}\right)$	0,0017	0,0720	0,0017	0,1795	0,1248
$k(\boldsymbol{x},\boldsymbol{y}) = \tanh(\langle \boldsymbol{x},\boldsymbol{y}\rangle + 1)$	$1 \cdot 10^{-10}$	0,1274	$2{,}05 \cdot 10^{-7}$	0,2562	0,1918

Jak můžeme vidět, nejlépe dopadl výpočet s námi používaným Gaussovým jádrem, následovalo jádro Laplaceovo a nejhůře ze tří testovaných dopadlo jádro tangens-hyperbolické.

Závěr

Cílů této práce bylo několik. V první řadě seznámit se s vybranými metodami strojového učení, konkrétně s metodami Kernel Ridge Regression a LASSO. K odvození a pochopení těchto metod bylo nejprve zapotřebí prozkoumání metody nejmenších čtverců. LASSO bylo zmíněno jen stručně, neboť největší důraz byl kladen na metodu Kernel Ridge Reggresion. Bylo provedeno její odvození, vysvětlení, dokázání některých vlastností a diskuze jejího možného použití.

Druhým cílem bylo seznámit se s konkrétními daty, na kterých bylo možné si teoretické poznatky z první části vyzkoušet. Těmito daty byly materiály vhodné k výrobě solárních panelů. Data pocházela ze soutěžě NOMAD 2018, která probíhala v roce 2018 na serveru Kaggle. Obsah dat, tedy informace popisující jednotlivé materiály, bylo nutné prozkoumat a pochopit za účelem možného vylepšování použitých metod strojového učení. Především šlo o dva předpovídané parametry, a to o formační energii a energii zakázaného pásu. Rovněž byla provedena jednoduchá vizualizace dat.

Posledním úkolem, neméně důležitým, bylo vyzkoušet si popsané metody strojového učení na rovněž zmíněných datech a pokusit se o zlepšení výsledků. K tomuto účelu byl vytvářen program v programovacím jazyce Matlab. Od jednoduchých výpočtů, které pracovaly pouze s původními dodanými informacemi, byla postupně předpověď zlepšována. Nejdůležitějším předpokladem pro dosažení dobrých výsledků byla práce se soubory, které obsahovaly prostorové souřadnice vždy všech atomů jedné krystalové buňky každého materiálu. Díky těmto informacím byly nakonec spočteny tzv. monogramy, které pomohly k vytvoření nových deskriptorů a také ke zlepšení výsledků. V měřítku soutěže by to znamenalo umístění v první pětadvacítce z více než 880ti účastníků.

Všechna výše komentovaná práce mě dovedla k dobrému pochopení zmíněných metod strojového učení. Největším přínosem však byla praktická aplikace na reálná data. Na té bylo možno pochopit teorii a vidět ji aplikovanou v praxi. Ne všechny pokusy o zlepšování výsledků vedly ke správnému cíli, nicméně z každého nezdařeného pokusu jsem si odnesl ponaučení do následující práce.

Do budoucna vidím v této oblasti mnoho příležitostí pro zlepšování předpovídání parametrů. Je zřejmé, že obě zmiňované energie velmi souvisí s chemickými vazbami mezi atomy krystalů a prostorovým uspořádáním atomů obecně. Hledání vhodných deskriptorů, které by vycházely z geometrických informací o datech a také z chemických vlastností z nich vyvozených, jistě povede k více přesnějším předpovědím.

Literatura

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

[2] L. M. Ghiringhelli, J. Vybiral, E. Ahmetchik, R. Ouyang, S. V. Levchenko, C. Draxl, M. Scheffler: *Learning physical descriptors for materials science by compressed sensing*. New Journal of Physics 19, 2017, 023017.

[3] L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, M. Scheffler: *Big data of materials science - Critical role of the descriptor*. Phys. Rev. Lett. 114, 2015, 105503.

[4] H. V. Henderson, S. R. Searle: *On Deriving the Inverse of a Sum of Matrices*. SIAM Review, 23(1):53–60, January 1981

[5] HyperPhysics (2015). Band Theory of Solids [online]. [cit. 04.07.2020]. Dostupné z: `http://hyperphysics.phy-astr.gsu.edu/hbase/solids/band.html`

[6] Ambarish Jash (2020). Kernel Ridge Regression, MATLAB Central File Exchange. [online]. [cit. 04.07.2020]. Dostupné z: `https://www.mathworks.com/matlabcentral/fileexchange/27248-kernel-ridge-regression`

[7] Mapping DFT Energies to Zacros Input (2020). *Zacros - Home* [online]. [cit. 04.07.2020]. Dostupné z: `http://zacros.org/tutorials/10-tutorial-4-mapping-dft-energies-to-zacros-input?start=3`

[8] K. R. Mueller, S. Mika, G. Ratsch, K. Tsuda, B. Schoelkopf: *An introduction to kernel-based learning algorithms*. IEEE Transactions on Neural Networks, 12(2), 2001, 181-201.

[9] Nomad2018 Predicting Transparent Conductors | Kaggle. *Kaggle: Your Machine Learning and Data Science Community* [online]. Dostupné z: `https://www.kaggle.com/c/nomad2018-predict-transparent-conductors`

[10] W. Rudin: Fourier Analysis on Groups. University of Wisconsin: Interscience Publishers, 1962. ISBN 9780685204368.

[11] J. Stewart: *Positive Definite Functions and Generalizations, an Historical Survey*. The Rocky Mountain Journal of Mathematics, 6(3), 1976, 409-434.

[12] C. P. Sutton, L. M. Ghiringhelli, T. Yamamoto, Y. Lysogorskiy, L. M. Blumenthal, T. Hammerschmidt, J. Gołębiowski, X. Liu, A. Ziletti, M. Scheffler: *NOMAD 2018 Kaggle Competition: Solving Materials Science Challenges Through Crowd Sourcing*. arXiv: Materials Science, 2018, Sutton2018NOMAD2K

[13] J. Thickstun: *Mercer's Theorem*. University of Washington (2018). [online]. [cit. 04.07.2020]. Dostupné z: https://homes.cs.washington.edu/~thickstn/docs/mercer.pdf

[14] V. N. Vapnik: *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995. ISBN: 978-1-4757-2440-0

[15] M. Welling: *Kernel Ridge Regression* [online]. [cit. 04.07.2020]. Dostupné z: https://www.ics.uci.edu/~welling/classnotes/papers_class/Kernel-Ridge.pdf

[16] Ball-and-stick model of part of the crystal structure of corundum, $\alpha-Al_2O_3$ - Wikimedia Commons. [online]. Dostupné z: https://commons.wikimedia.org/wiki/File:Corundum-3D-balls.png