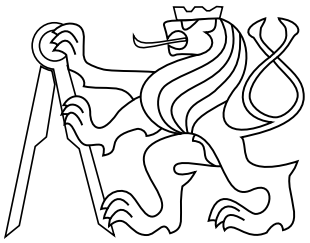
Bakalářská práce

Klasifikace EEG s využitím metod inkrementálního učení

Petr Husák



květen 2014

Vedoucí práce: Ing. Václav Gerla, Ph.D.

České vysoké učení technické v Praze

Fakulta elektrotechnická, Katedra kybernetiky

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Petr H u s á k

Studijní program: Otevřená informatika (bakalářský)

Obor: Informatika a počítačové vědy

Název tématu: Klasifikace EEG s využitím metod inkrementálního učení

Pokyny pro vypracování:

Student naváže na bakalářskou práci Bc. Mateje Murgaše [1], který sestavil dvě metody inkrementálního učení a použil je nad spánkovými EEG záznamy. Jeho práci rozšíří následujícím způsobem:

1. Původní metody umožnují inkrementální trénování klasifikátoru, ovšem samotná klasifikace se provádí v každém kroku nad celým datasetem, což může být u dlouhodobých EEG záznamů nevyhovující. Student si vybere jednu z již implementovaných metod a optimalizuje ji, a to tak, aby umožnovala efektivnější/rychlejší klasifikaci při použití nad většími datasety.
2. Původní metody byly testovány nad záznamy, které byly rozděleny na segmenty konstantní délky, totožné pro všechny zpracovávané kanály. Student upraví vybranou klasifikační metodu, aby ji bylo možné použít nad adaptivně segmentovanými EEG/PSG záznamy.
3. Student se seznámí se způsoby měření EEG signálu a s vedoucím práce provedou v laboratoři skupiny BioDat testovací měření. Zaměří se na úlohu detekce očních mrkání a svalových artefaktů v EEG záznamu.
4. Nad naměřenými daty student otestuje jím navržené algoritmy a provede statistické zhodnocení výsledků.

Seznam odborné literatury:

[1] Murgaš M.: Inkrementální učení v úloze klasifikace EEG signálu, Bakalářská práce, ČVUT-FEL,Praha, 2013.
[2] Gerla V.: Automated analysis of long-term EEG signals, Ph.D. thesis, ČVUT-FEL, Praha, 2012. URL http://bio.felk.cvut.cz/psglab/disertace/disertace-2012-02-29.pdf
[3] Geng X. and Smith-Miles K.: Incremental Learning. Stan Z. Li ed.: Encyclopedia of Biometrics, Springer, NY, USA, 2009.

Vedoucí bakalářské práce: Ing. Václav Gerla, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Václavu Gerlovi, Ph.D za ochotu a cenné připomínky k této práci.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne 23. 5. 2014 .....

Petr Husák

Abstrakt

Cílem práce je přizpůsobit a optimalizovat adaptivně nebo konstantně segmentovaný EEG záznam pro použití inkrementálního učení na dlouhodobých záznamech. Jsou použity již navržené algoritmy pro segmentaci a výpočet příznaků z PSGLab toolboxu [1]. Snahou je sjednotit matematický popis příznaků získaných adaptivní a konstantní segmentací a dále příznaky co nejvíce zredukovat pro urychlení výpočtu shlukování a klasifikace nejbližším sousedem. Při inkrementálním učení docházelo k neefektivnímu přehodnocování celého záznamu po každém nově přidaném segmentu do trénovací množiny. Jako řešení byly navrženy dvě metody jak se tomu vyvarovat a přehodnocovat jen potřebné části.

Klíčová slova

EEG; PCA; k-means; KNN; Voronovy diagramy; klasifikace; inkrementální; online; adaptivní; učení

Abstrakt

The aim of this thesis is to adjust and optimize adaptive or constant segmented long-term signals in task of incremental learning. Already proposed algorithms are used for segmentation and feature extraction implemented in PSGLab toolbox [1]. The goal is to unify the mathematical description of features which are obtained from the constant or adaptive segments. This feature space is still huge therefore it is reduced due to the time and space complexity of clustering and nearest neighbor classification. In the task of incremental learning the classification of the whole signal was performed after every addition of a new segment into train set. As a result of this two methods were designed to classify only necessary parts of signals.

Keywords

EEG; PCA; k-means; KNN; Voronoi diagrams; classification; incremental; online; learning

Obsah

Zkratky

PCA	Analýza hlavních komponent (en: Principal component analysis)
KNN	Klasifikátor hodnotící podle k nejbližších sousedů
KD strom	K-dimenzionální strom
DT	Delaunayova triangulace
VD	Voronův diagram
QS	Klidný spánek
AS	Aktivní spánek

1 Úvod

Automatická klasifikace EEG záznamů je v klinické praxi nespolehlivá a lékaři ji nepoužívají. Zlepšením by byl nástroj k ohodnocování záznamu, který by nabízel lékařům klasifikované stavy, které by byly buď schvalovány, nebo měněny. Podle toho by se klasifikátor přeučoval a nabízel stále přesnější výsledky, dokud by nebyl celý záznam ohodnocen.

Tento způsob se nazývá inkrementální učení. Tématem práce není předzpracování EEG záznamu, ale je počítáno s již konstantně nebo adaptivně segmentovanými a parametricky popsanými záznamy získanými z PSBLab toolboxu [1]. Prvním krokem je snížit počet získaných parametrů u dlouhodobých záznamu a zapsat je do struktury pro použití klasifikátoru. Klasifikace není zaměřena na konkrétní typ záznamů a parametry budou redukovány obecně metodou analýzy hlavních komponent. Snížením dimenze příznaků se urychlí klasifikace nejbližším sousedem.

Pro získání přesného klasifikátoru už v počátku klasifikace, bude použito shlukování k-means. Časové náročný proces představuje klasifikace záznamu po každém inkrementálním kroku. Tento proces lze zefektivnit a hodnotit pouze části, které jsou aktuálně potřebné, nebo dokonce jen ty, které mohou být inkrementálním krokem ovlivněny. Implementace metod bude provedena v prostředí MATLAB.

Podobnými tématy se zabývají bakalářské práce *Incremental Learning in the Task of EEG Signal Classification* [2], která porovnává klasifikátor nejbližšího souseda a SVM, a *Analýza epileptických EEG signálů* [3], která se zaměřuje na sestavení sady příznaků pro automatické hodnocení epileptických záznamů. Hlavní inspirací pro zpracování dlouhodobých záznamů byla disertační práce *Automated Analysis of Long-Term EEG Signals* [4].

2 EEG signál

2.1 Získání EEG signálu

2.1.1 Biologický význam

Elektroencefalografie je metoda vyšetření mozkové aktivity. Lidský mozek se skládá z neuronů, které tvoří základní stavební jednotky. Mezi neurony existuje membránový potenciál, který má hodnoty okolo $-70\mu V$. Ke vzruchu dochází, pokud se překročí prahový potenciál (okolo $-55\mu V$). Nastane tzv. akční potenciál, kdy hodnota membránového potenciálu vystřelí k $100\mu V$ a rychle klesne. Pro tento děj má každý neuron dvě důležité části axon a dendrit. Axon slouží k vyvedení akčního potenciálu, tedy informaci vysílá. Dendrit vstupní informaci přijímá. Každý neuron dostává elektrický impuls od tisíců až statisíců jiných neuronů, které zpracuje a vysílá dál do neuronové sítě [4].
Mozek vykazuje širokou škálu aktivit během normálních i patologických stavů. Z normálních stavů jsou to stavy fyziologické jako spánek, bdělost, oční pohyby nebo svalové artefakty a mentální například vztek, štěstí, stres, ... Mezi patologické patří epileptické záchvaty, Alzheimerova choroba a mnoho dalších [5]. Cílem elektroencefalografie je některé takové stavy zachytit a diagnostikovat.

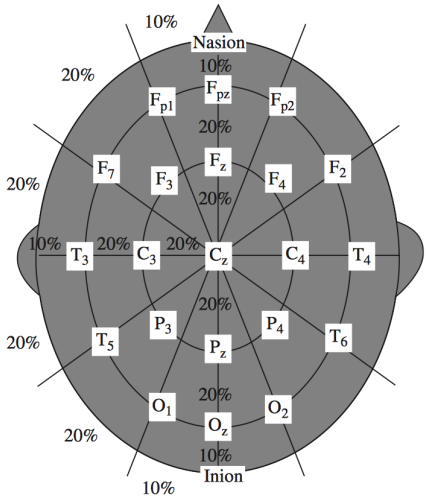
2.1.2 Měření

Přístroj k měření EEG se nazývá encefalogram. Měření probíhá tak, že je pacientovi na hlavu umístěná čepice s elektrodami, které snímají elektrický potenciál neuronů ve svém okolí. Elektrody jsou obvykle na čepici rozmístěné podle systému 10-20. každá elektroda nese písmenné a číselné označení. Písmeno znamená

- F = frontální (čelní),
- F_p = frontopolární,
- C = centrální,

- O = okcipitální (týlový),
- P = paretální (temenní),
- T = temporální (spánkový).

A číselný index označuje na jaké hemisféře je elektroda umístěná, lichý značí levou hemisféru, sudý pravou a písmeno z nepárové elektrody [6, 4].



Obrázek 1 10-20 systém rozmístěním 21 elektrod [6].

Naměřený signál lze rozložit do pěti frekvenčních pásem.

- Delta (méně než 4Hz) je přítomna v hlubokém spánku.
- Theta (4Hz - 7Hz) je minimální v bdělosti dospělých, objevuje se však v ospalosti. Je velmi častá u dětí.
- Alpha (7Hz - 12Hz) přítomna při relaxaci a zavřených očích.
- Beta (12Hz - 30Hz) aktivita je typická pro koncentraci, myšlení, řešení problémů. Vysoké frekvence můžou značit stavy paniky.
- Gamma (30Hz a více) aktivita se objevuje jen zřídka. Uvádí se, že přítomnost značí stavy meditace, ale může být známkou nějaké choroby [4].

Při měření záznamu dochází k rušení, kterému se říká technické artefakty. Patří mezi ně například rušení elektrickou sítí (50Hz), jinými elektrickými zařízeními nebo jinými okolními signály.

Ze stany pacienta dochází také k nežádoucím jevům, které mohou zakrývat zkoumaný jev. Jsou to biologické artefakty. Mezi ně patří svalové pohyby, oční pohyby nebo na-

příklad dýchání a rytmus srdce [4].

2.2 Předzpracování

Cílem předzpracování signálu je odstranění nežádoucích jevů jako jsou artefakty. Typickým příkladem je odstranění 50Hz rušení elektrickou sítí.
Pro dlouhé signály vzorkované vysokou vzorkovací frekvencí vznikají objemná data, která jsou náročná na další zpracování. Proto se provádí převzorkování, tj. změna vzorkovací frekvence. Jejím snížením se získají menší data.

2.3 Segmentace

Segmentace je krok předzpracování, při kterém se signál rozdělí na menší úseky, které se dále zpracovávají. Segmentace je důležitá pro datový popis segmentů a pro automatickou klasifikaci, kdy se segmenty klasifikují. [4]

2.3.1 Konstantní segmentace

Konstantní segmentace rozdělí signál na úseky stejné délky. Výhodou je rychlost a vždy stejný časový úsek pro každou elektrodu. Nevýhodou příliš těžkopádné rozdělení a segmenty jsou často nepoužitelné, protože obsahují více aktivit, které je třeba rozpoznávat zvlášť.
V této práci jsou používané segmenty s délkou 30s v záznamech spánku dospělých.

2.3.2 Adaptivní segmentace

Adaptivní segmentace rozděluje signál na úseky proměnné délky. Je mnoho způsobů na výpočet. Adaptivní segmenty v této práci jsou získána metodou dvou po sobě jdoucích oken, které počítají a porovnávají charakteristiky signálů. Metoda je popsána v [4].
Jsou zde používány adaptivně nasegmentované signály spánku novorozenců a artefakty.

2.4 Parametrizace signálu

Dalším krokem ve zpracování signálu je datový popis jednotlivých segmentů. Cílem je co nejpřesněji popsat celý segment pro klasifikátor, avšak příliš detailní popis může znamenat velkou paměťovou i časovou náročnost dalších metod včetně klasifikátoru.

Vypočtená hodnota se nazývá příznak. Proces vybírání lze rozdělit na extrakci příznaků a selekci příznaků. [4, 3]

2.4.1 Extrakce příznaků

Extrakce příznaků je výpočet hodnot přímo ze signálu. Ze statistických hodnot jsou to průměr, směrodatná odchylka, minimální a maximální hodnota, šikmost, špičatost a medián. Dále se počítají analýzy založené na derivacích, Hjortových parametrech, waveletové transformaci, frekvenční analýze, entropii a dalších [4].

2.4.2 Selekce příznaků

Do selekce spadá výběr příznaků. Použití mnoha příznaků by bylo nadbytečné a neefektivní, je nutné prostor příznaků redukovat. Mezi takové metody patří například *info gain attribute evaluation* nebo χ^2 *attribute evaluation* [2]. V této práci byla zvolena analýza hlavních komponent, které je věnována další část kapitoly.

2.4.3 Matematický popis segmentů

Budeme uvažovat, že chceme celý datový popis záznamu zapsat do matice. V případě konstantní segmentace nevzniká žádný problém. Celý záznam je rozdělen na stejně velké díly, ze kterých jsou spočítány parametry, takže lze uvažovat jeden segment napříč všemi kanály a parametry zapsat do vektoru matice. Výsledná matice bude mít ve sloupcích segmenty a v řádkách příznaky všech kanálů.
V případě adaptivní segmentace je situace jiná. Segmenty napříč kanálů nemají stejnou velikost a nelze jejich parametry zapsat do jednoho vektoru sloupce matice. Celý záznam je možné segmentovat konstantně s velikostí menší než je nejmenší adaptivní segment. Tyto konstantní segmenty budou mít hodnoty jejich nadřazených adaptivních segmentů. V případě rozmezí dvou adaptivních lze rozhodnout podle toho, do jaké části konstantní segment spadá více. Tímto vznikne matice jako v případě konstantně segmentovaného záznamu. V matici budou úseky vektorů vedle sebe totožné. Lze je odstranit jedním průchodem matice a zůstanou jen vektory reprezentující maximální adaptivní segmenty napříč všech kanálů.

2.5 Analýza hlavních komponent

Analýza hlavních komponent (en: Principal component analysis) je statistická metoda, která slouží ke snížení dimenze prostoru a korelace dat. Transformací původní sady příznaků vznikne nová méně korelovaná sada. Uplatnění PCA je například v extrakci příznaků, vizualizaci nebo kompresi. [7]
Cílem metody je získat uspořádanou bázi ortogonálních vektorů, které jsou seřazeny podle rozptylu. Jednotlivé vektory nové báze se nazývají komponenty. První komponenta vznikne jako vektor ve směru největšího rozptylu dat, druhá komponenta, kolmá na první, má směr dalšího největšího rozptylu. Takto se vytvoří báze celého prostoru, přičemž pro redukci dat do m-rozměrného prostoru stačí využít prvních m komponent. [5]

2.5.1 Výpočet pomocí vlastních čísel a vektorů

Mějme množinu $\mathbf{X} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^n, i = 1, \ldots, k\}$, kde k je počet pozorování a vektory $\mathbf{x}$ obsahují naměřené příznaky. Množinu je potřeba posunout do počátku soustavy.

$$\mu = \frac{1}{k} \sum_{i=1}^{k} \mathbf{x}_i \tag{1}$$

$$\bar{\mathbf{X}} = \{\mathbf{x}_i - \mu | \mathbf{x}_i \in \mathbb{R}^n, i = 1, \ldots, k\} \tag{2}$$

Sestavíme matici $\bar{\mathbf{X}}$, která má ve sloupcích vycentrované vektory $\bar{\mathbf{x}}_i$. Matice obsahuje v řádcích stejné příznaky v různých realizacích a ve sloupcích všechny příznaky.

$$\bar{\mathbf{X}} = \begin{bmatrix} \bar{\mathbf{x}}_1 & \bar{\mathbf{x}}_2 & \ldots & \bar{\mathbf{x}}_k \end{bmatrix} \tag{3}$$

Spočítáme matici kovariance, která je čtvercová a pozitivně semidefinitní. Matice kovariance má na hlavní diagonále rozptyly jednotlivých příznaků. Velké hodnoty odpovídají důležitým datům, malé naopak šumu. Na ostatních místech matice je kovariance každých dvou příznaků. Velké číslo znamená nadbytečnost.

$$\mathbf{C}_\mathbf{X} = \frac{1}{k} \bar{\mathbf{X}} \bar{\mathbf{X}}^T \tag{4}$$

Optimální kovariantní matice by byla diagonální matice $\mathbf{C_Y}$. Ukáže se [8], že existuje ortogonální projekční matice $\mathbf{P}$, tak že

$$\mathbf{Y} = \mathbf{P}\bar{\mathbf{X}}, \qquad \mathbf{C_Y} = \frac{1}{k}\mathbf{Y}\mathbf{Y}^T. \tag{5}$$

Řádky matice $\mathbf{P}$ jsou vlastní vektory matice $\bar{\mathbf{X}}\bar{\mathbf{X}}^T$, které se seřadily podle jim příslušným vlastním číslům sestupně. Jsou to seřazené hlavní komponenty PCA. [5, 8, 7]
Redukovaná data do dimenze m, $W = \{\mathbf{w}_i | \mathbf{w}_i \in \mathbb{R}^m, i = 1, ..., k\}$ se vypočítají

$$\mathbf{W} = \bar{\mathbf{P}}\bar{\mathbf{X}},$$

kde $\bar{\mathbf{P}}$ je m řádků matice $\mathbf{P}$.

2.5.2 Volba vhodného počtu komponent

Vhodný počet komponent pro výsledný prostor může napovědět rozptyl jednotlivých komponent. Rozptyly jsou vlastní čísla matice kovariance a jejich součet je celkový rozptyl dat. Pokud je seřadíme sestupně, stačí vzít prvních m komponent pro rozptyl $r = \sum_{i=1}^{k} \lambda_i$. V praxi obvykle stačí využít 65%-90% celkového rozptylu. [9]

3 Inkrementální učení

Inkrementální učení je metoda strojového učení, nazývána také aktivní učení nebo online učení. Dokáže změnit svůj model, jakmile se objeví nový testovací vzorek. Hlavní rozdíl od tradičního učení je, že na začátku nemá žádnou trénovací sadu a nové vzorky se objevují průběžně. Toto učení je mnohem přirozenější než učení tradiční. Málokdy jsou hned na začátku všechny informace dostupné. K inkrementálnímu učení vedou i důvody jako například příliš velká trénovací množina, která se buď nevejde do paměti, nebo by výpočet byl příliš složitý. Nehledě na to, kdyby se později objevil nový vzorek, celý složitý proces by se musel opakovat.[10]
Byly stanoveny tři kritéria pro inkrementální algoritmus [10].

- Stabilita - klasifikace na testovací sadě by se neměla chovat příliš divoce při každém inkrementálním kroku.
- Zlepšování - klasifikace na testovací sadě by se měla s přibývajícímu vzorky v trénovací sadě zlepšovat.
- Obnovitelnost - algoritmus by se měl umět dokázat vrátit do stavu před přidáním nového vzorku, pokud se zjistí, že klasifikace se zhoršila.

Typickým příkladem inkrementálního algoritmu je metoda nejbližšího souseda, které je věnována kapitola 4. Další metodou, kterou lze použít je SVM. Dalším pohledem na inkrementální učení je, co nového model může očekávat [10].

- Učení pouze nových příkladů a tedy zpřesňování klasifikátoru.
- Učení nových tříd. Mezi trénovacími vzorky se může objevit zcela nová třída, které je potřeba model přizpůsobit.
- Učení nových příznaků. V průběhu se mohou objevit nové příznaky.

V řešené úloze mohou nastat první dvě skupiny. Pro nejbližšího souseda není potřeba model nijak přizpůsobovat novým třídám.
V úloze se předpokládá, že signál EEG bude hodnotit reálná osoba, tudíž vzorky se

budou hodnotit postupně a model se bude zvětšovat. Nemusí však platit, že se úseky signálu musí hodnotit postupně. Je možné pro zlepšení výsledku nejdříve vytvořit základní model, který získá o datech lepší přehled. To znamená najít shluky a jejich vybrané zástupce nechat ohodnotit. Díky tomuto modelu se oklasifikuje celý signál a hodnotící osoba může pak signál procházet postupně a potvrzovat či měnit ohodnocené segmenty, přičemž model se bude inkrementálně rozšiřovat.

3.1 Počáteční stavba modelu

Na začátku nemáme o datech žádnou představu a nabízí se proto několik možností, jak začít hodnotit. Čistě z inkrementálního pohledu učení bychom mohli začít postupně hodnotit signál, tím se ale může stát, že 50% signálu budeme hodnotit jednu třídu a o dalších, které v signálu mohou být, nebudeme mít žádné informace. Jinou jednoduchou možností je vybírat vzorky náhodně. Tato metoda má své výhody v náhodném prohledávání prostoru [11].
Další způsob je získat o datech nějaký přehled pomocí metod učení bez učitele.

3.1.1 K-means

Algoritmus K-means je nehierarchický algoritmus, který data rozdělí do předem zadaných K shluků, tak že každý z n vzorků, patřící do shluku k má kratší vzdálenost ke svému centroidu $\mathbf{c}_k$, než ke všem ostatním. K-means minimalizuje sumu

$$\arg\min_{j} \sum_{i=1}^{n} \sum_{k=1}^{K} ||\mathbf{x}_i - \mathbf{c}_k||^2$$

Při předem neznámých datech je těžké odhadnout počáteční počet shluků. Jedním způsobem pro odhad jsou hodnoty silhouette grafu, které určují, jak blízko jsou si body ve stejných shlucích v porovnání s ostatními shluky. Definujeme hodnotu $s(x_i)$ pro každý bod x_i, $i = 1, ..., n$. Máme K shluků, označených $\mathbf{C}_k$, $k = 1, ..., K$ a bod $p \in \mathbf{C}_a$.

$$a(p) = \text{průměrná vzdálenost bodu } p \text{ a bodů ve stejném shluku } \mathbf{C}_a,$$

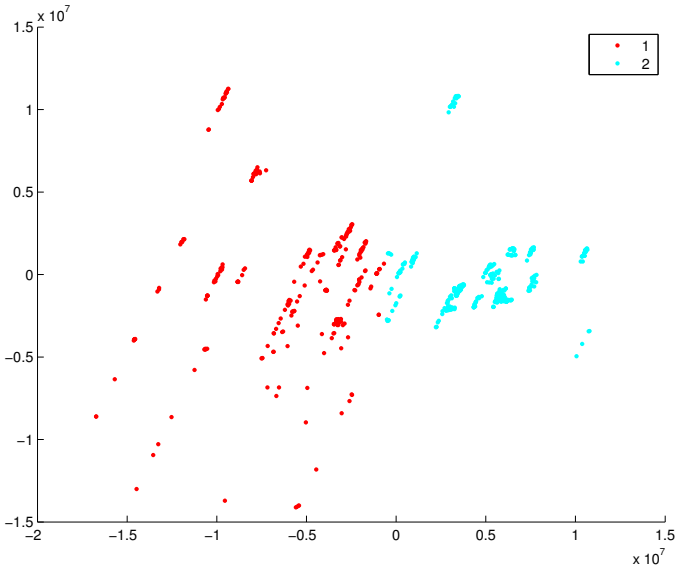
$$d(p, \mathbf{C}_l) = \text{průměrná vzdálenost bodu } p \text{ a bodů jiného shluku } \mathbf{C}_{l \neq a},$$

$$b(p) = \min_{\mathbf{C}_{j, j \neq a}} d(p, \mathbf{C}_j).$$

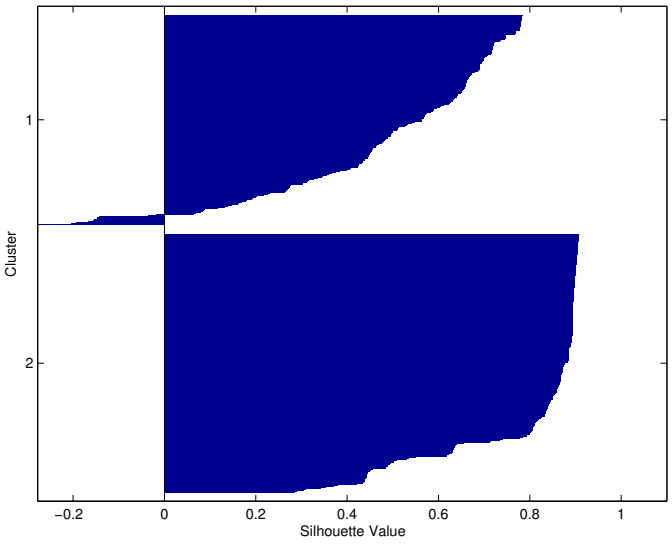
Hodnota

$$s(p) = \frac{b(p) - a(p)}{\max\{a(p), b(p)\}}$$

nabývá rozpětí od +1 do -1. Vysoká hodnota znamená, že bod je hodně vzdálen od sousedních shluků, a nízká hodnota znamená, že bod bude pravděpodobně zařazen špatně. Průměrem hodnot všech bodů lze relativně posoudit, správnost volby počtu shluků [12, 13].



a)



b)

Obrázek 2 Výsledek shlukové analýzy pro dvě třídy ve dvourozměrném prostoru (A) a odpovídající sillhouette graf (B) [14].

Algorithm 1 Calculate K-Means

Vstup:

$X = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^n$ {množina příznaků}

$K \in \mathbb{R}$ {počet shluků}

Výstup:

$\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ {množina středů shluků}

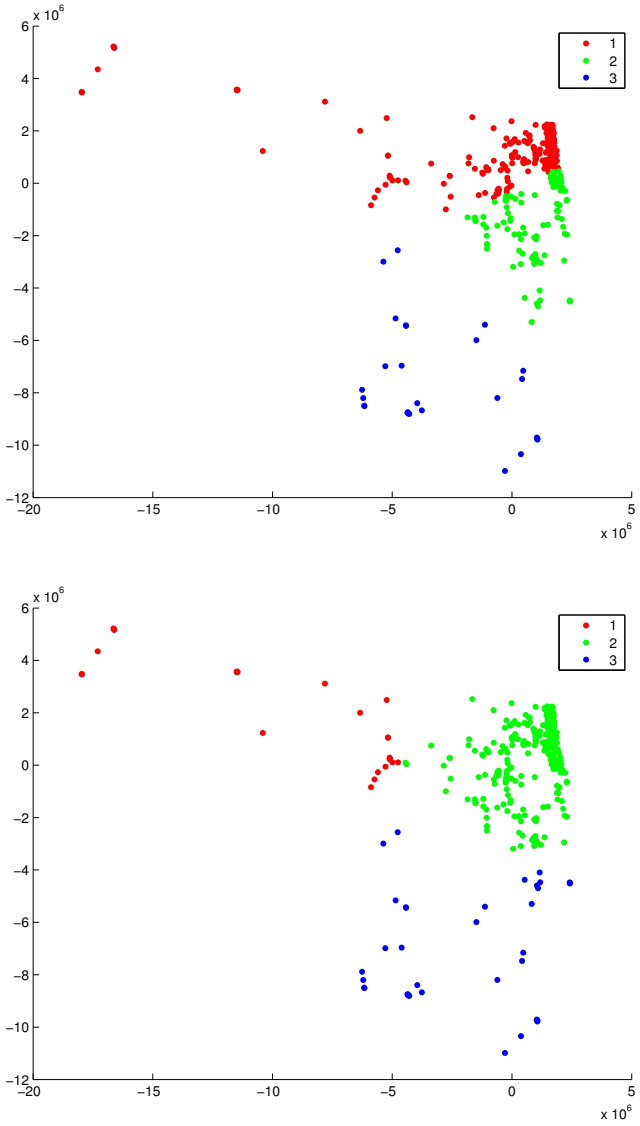
$\{X_k\}_{j=1}^K\}$ {množiny příznaků ve shlucích}

1: Inicialializovat $(\mathbf{c}_i)_{i=1}^k$

2: Klasifikovat vzorky $\mathbf{x}_i$ podle nejbližšího $\mathbf{c}_k$
$X_k = \{\mathbf{x} \in X : \forall j, ||\mathbf{x} - \mathbf{c}_k||_2^2 \leq ||\mathbf{x} - \mathbf{c}_j||_2^2\}$

3: Přepočítat středy shluků, tak že
$\mathbf{c}_k = \arg\min_c \sum_{\mathbf{x} \in X_k} ||\mathbf{x} - \mathbf{c}||_2^2 = \frac{1}{|\mathbf{X}_k|} \sum_{\mathbf{x} \in \mathbf{X}_k} \mathbf{x}$

4: Ukončit, pokud $X_k^{t+1} = X_k^{t+1}, \forall k$; jinak jdi na 2.



Obrázek 3 K-means se stejným počtem shluků a různými výsledky.

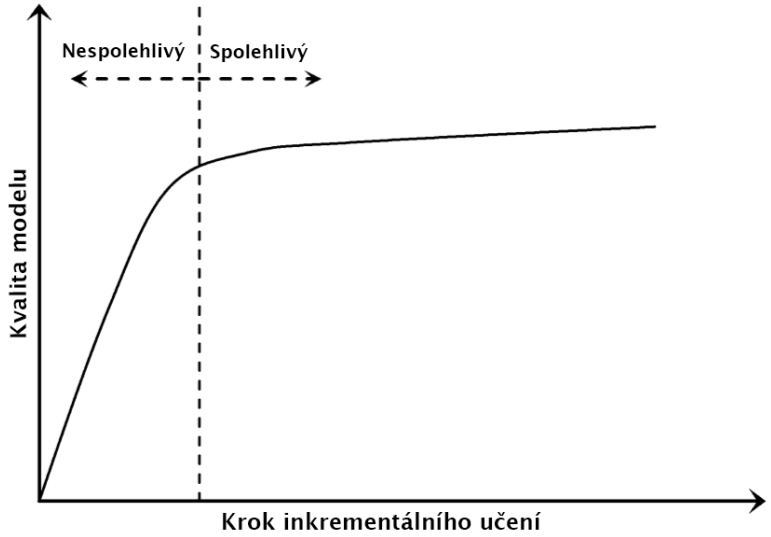
Algoritmus konverguje po t krocích. Jeho časová složitost je $O(ndct)$, kde n je počet vzorků, d je dimenze příznaků, c je počet shluků a t je počet iterací algoritmu [5]. Jelikož je algoritmus inicializován náhodně, může se stát, že konverguje do lokálního minima, nikoli globálního, viz Obr. 2. Jednoduchým řešením je algoritmus s náhodnou inicializací spustit několikrát a porovnat výsledky vzdáleností [6, 15]. Předpokládáme eukleidovskou metriku, algoritmus lze jinak zobecnit i pro jiné druhy metrik.

3.2 Rozšiřování modelu

Model je následně nutné zpřesňovat, dokud se nestane dostatečně spolehlivým nebo signál nebude ohodnocen celý. Postupy ohodnocování signálu mohou být následující.

- Hodnotící osoba bude signál procházet podle vlastní vůle, tedy s největší pravděpodobností signál bude hodnotit postupně.
- Model bude sám určovat podmíněnou pravděpodobnost každého vzorku k dané třídě a dle toho nabízet vzorky k ohodnocení.

Ukázkou kvality inkrementálního učení je učící křivka, která znázorňuje zlepšování kvality modelu, tedy správnosti klasifikace celého signálu, v závislosti na počtu naučených příkladů, tedy zvětšování trénovací množiny. Toto lze použít jako kritérium o rozhodnutí, zda je model dostatečně spolehlivý.



Obrázek 4 Inkrementální učící křivka. Přeloženo z [10].

Na začátku je model nespolehlivý a v ideálním případě by se po každém inkrementálním kroku měla klasifikace zpřesnit. Učící křivka by měla být rostoucí a konvergovat ke 100% přesnosti. V Experiment na EEG záznamechu bude vidět, že jeden vzorek může celý model ovlivnit i velmi negativně.

4 Klasifikace metodou nejbližšího souseda

Metoda k nejbližších sousedů je neparametrická metoda strojového učení. Její vstup je trénovací množina vektorů příznaků, na které se metoda naučí a následně ke každému testovacímu vzorku najde k nejbližších sousedů na základě metriky. Přestože metoda je z hlediska klasifikátoru velmi jednoduchá, při velké sadě dat vznikají problémy s paměťovou a výpočetní náročnosti.

Velkou výhodou je, že algoritmus se dokáže přeučovat bez jakékoli změny modelu, tedy je velmi vhodný pro inkrementální učení. Každý nově oklasifikovaný vzorek se může okamžitě přidat do trénovací sady a testovací data se podle něj začnou okamžitě klasifikovat.

V této kapitole se budeme zabývat jaké existují optimalizace a jak klasifikaci urychlit z hlediska inkrementálního učení.

4.1 Vlastnosti klasifikátoru

Rozdíl metody nejbližších sousedů od parametrických metod je, že nepotřebuje znát o datech pravděpodobností model, je ale nutné mít větší množinu trénovacích dat.

Metoda vytváří kolem testovaného vzorku zvětšující se hyperkouli dokud nepohltí k trénovacích vzorků. Pokud máme dvě třídy volíme lichý počet nejbližších sousedů, pokud klasifikujeme více tříd, mohou nastat situace, kdy nelze jednoznačně rozhodnout, kterou třídu zvolit.

Výhodou metody je jednoduchá implementace a používá se tudíž jako referenční klasifikátor. Chyba klasifikace vychází v aplikacích podobně jako chyba neuronových sítí. Nevýhoda je, že neexistuje žádná generalizace chyby. Na trénovací množině vychází 0 a Vapnik - Červoněnkisova dimenze je nekonečno, chybu na trénovací množině nelze vyjádřit. Asymptotickou chybu lze však odhadnout pomocí Bayesovské chyby a to

$$P^* < P < 2P^*, \tag{6}$$

kde P^* je Bayesovská chyba a $P = \lim_{n\to\infty} P_n$ je asymptotická chyba NN klasifikátoru při n vzorcích v trénovací množině.

Časová náročnost naivního algoritmu, který bude počítat všechny vzdálenosti k jednomu vzorku a bude si pamatovat jen nejkratší je $O(dn)$, kde d je dimenze a n je počet vzorků. Při zvětšování trénovací sady vzniká i velká paměťová náročnost, která se ale dá řešit kondenzací trénovací množiny.

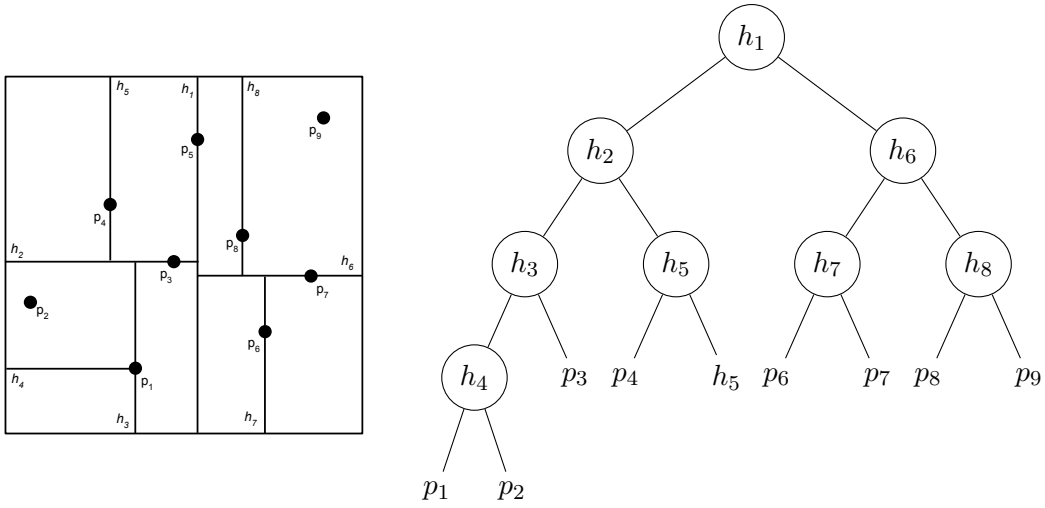
Metoda je závislá na metrice a je tedy nutno data normalizovat. [16, 17]

4.2 Optimalizace na trénovací sadě

Jelikož klasifikujeme na základě metriky stává se z metody problém výpočetní geometrie a optimalizace se budou týkat reprezentace a zjednodušování n-rozměrného prostoru.

4.2.1 K-dimenzionální stromy

K-dimenzionální stromy mají za úkol snížit časovou náročnost při hledání nejbližších sousedů v prostoru. Data trénovací množiny se setřídí podle jedné souřadnice a podle mediánu nebo střední hodnoty se data rozdělí na dvě podmnožiny. Každá z nich se zase setřídí a rozdělí. Postup se opakuje dokud nevzniknou buňky s právě jedním bodem. Na Obr. 5 je vidět jak je prostor rozdělen ve 2D. [17]

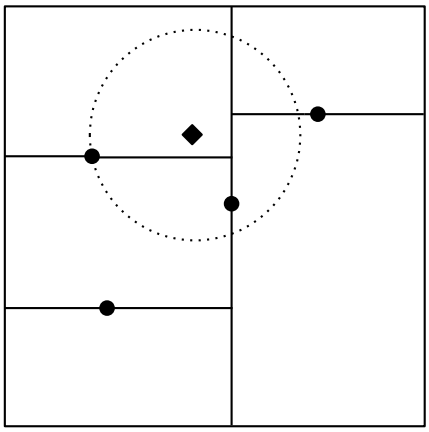


Obrázek 5 Ukázka rozdělení 2D prostoru (medián) a jeho reprezentace v kd stromu.

Časová složitost sestavení stromu použitím efektivních algoritmů je $O(n \log n)$.

Hledání nejbližšího souseda ve stromu probíhá tak, že se strom projde a nalezne kon-

krétní buňku. Její přidružený bod, ale ještě nemusí být nejbližší soused, viz Obr. 6. Změří se vzdálenosti k přidruženému bodu a všem hranicím buňky. Jestliže některé hranice jsou blíže než bod, musí se prohledat i buňky za hranicí [17]. Podobně lze hledat i více nejbližších sousedů [18].



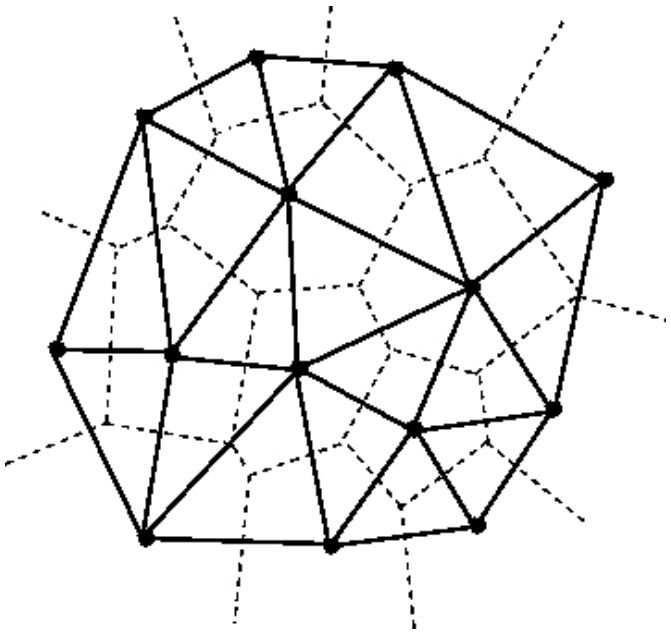
Obrázek 6 Vzdálenost k přidruženému bodu je větší než kolmá vzdálenost k dělícím přímkám, je potřeba prohledat i sousední buňky.

Časová složitost nalezení K nejbližších sousedů je $O(K \log n)$. S růstem dimenze prostoru dochází k nutnosti prohledávat více podprostorů a algoritmus přestává být o tolik efektivnější než lineární prohledávání. Ukázky a popis algoritmu [19, 20, 21].

4.2.2 Delaunayovy triangulace a Voronovy diagramy

Voronův diagram rozdělí prostor podle nejbližšího souseda. Vznikne síť buněk, kde v každé buňce budou body klasifikovány podle bodu, jemuž je buňka přiřazená. V rovině vznikne síť konvexních polygonů, ve 3D konvexní mnohostěny.
Duálním grafem jsou Delaunayovy triangulace. Vztahy mezi těmito grafy jsou

- Body středů VD jsou vrcholy DT.
- Dva body středů VD mají mezi sebou hranu v DT právě tehdy, když sdílejí Voronovu hranu.
- Střed kružnice opsané DT je Voronův vrchol.



Obrázek 7 Voronův diagram (čárkovaně) a Delaunayovy triangulace (spojitá čára) [22].

Nevýhodou VD je složitost výpočtu $O(n \log n + n^{\frac{d}{2}})$ [17].

Metodou jak snížit náročnost prohledávání nejbližších sousedů je redukovat trénovací množinu. Jednou z možností je z trénovací množiny odstranit takové vzorky, které jsou obklopeny vzorky stejných. Algoritmus pro redukci trénovací množiny předpokládá konstrukci Voronových diagramů.

Algorithm 2 Editace nejbližšího souseda

1: Inicialializovat $j = 0$, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ trénovací množina
2: Konstrukce Voronoiova diagramu
3: **for all** $\mathbf{x} \in X$ **do**
4: Najít všechny Voronoiovy sousedy vzorku $\mathbf{x}$
5: **if** Všichni sousedi mají stejnou třídu **then**
6: Označit $\mathbf{x}$
7: **end if**
8: **end for**
9: Odstranit všechny označené vzorky z trénovací množiny.

Časová složitost algoritmu je $O(d^3 n^{\lfloor d/2 \rfloor} \ln n)$. V případě inkrementálního přidávání vzorků, lze při každém přidání nového vzorku otestovat potencionální Voronovy sousedy [16].

4.3 Optimalizace na testovací sadě

4.3.1 Pomocí Voronových diagramů

Při inkrementálním zvětšování trénovací sady je třeba pokaždé překlasifikovat celou testovací sadu. Změna v trénovací množině, ale neovlivní celou testovací sadu. Je nutné překlasifikovat jen určitou množinu, která může být změnou ovlivněna. K tomu lze využít VD a to tak, že možné změny klasifikace postihnou jen vzorky, které jsou v podprostoru tvořeném novou Voronovou buňkou a jejími sousedními buňkami. Kvůli časové i paměťové náročnosti je metoda ve vyšších dimenzí téměř nepoužitelná (dle manuálu MATLABu dimenze vyšší než 6) a bylo by rychlejší celou testovací sadu oklasifikovat znovu. Metoda je dobře využitelná pro dvou a třírozměrný prostor, kde existují velmi efektivní algoritmy pro výpočet.

4.3.2 Klasifikace potřebné části signálu

Během inkrementálního učení nebude potřeba mít najednou ohodnocený celý záznam. Bylo by proto neefektivní ho po každém kroku celý znovu hodnotit. Lze využít toho, že signál bude hodnotit osoba, která bude mít v čase zobrazený pouze omezený úsek záznamu. Bude se tedy hodnotit právě zobrazený úsek a s posouváním zobrazeného úseku se budou klasifikovat nové segmenty.

5 Implementace

V této části bude stručně popsaná implementace aplikace. Některé části jsou přizpůsobený spíše jen pro získání statistických dat a grafů. Část by mohla být v budoucnu použita v PSGLabu.

5.1 Načtení dat

Na začátku se načtou příznaky ze souboru `arff`, který je popsán zde [23] a jeho ukázka Obr. 8. Je to textový soubor rozdělený do dvou hlavních částí. V první části obsahuje název uvedený klíčovým slovem *@RELATION* a seznam příznaků, uvedených *@ATTRIBUTE*, formátu *@ATTRIBUTE název_atributu datový_typ*. Druhá část začíná klíčovým slovem *@DATA* a na dalším řádku obsahuje matici dat, kde řádky odpovídají segmentům a sloupce příznakům.

```
@RELATION FEATURES

@ATTRIBUTE ch_FP1---min_value REAL
@ATTRIBUTE ch_FP1---max_value REAL
@ATTRIBUTE ch_FP1---skewness REAL
...
@ATTRIBUTE Class {1, 2}

@DATA
-25.000 39.060 0.450 ... 1
-17.970 29.685 0.435 ... 1
-10.940 20.310 0.420 ... 1
-25.000 21.095 0.164 ... 2
.
.
.
```

Obrázek 8 Ukázka souboru `arff`.

Funkce `read_arff.m` načte soubor `arff` a výstupem je struktura, obsahující 3 proměnné.

Tabulka 1 Funkce k načtení příznaků `read_arff(X)`.

`[P] = read_arff(X)`	
X	Adresa souboru typu `arff`
P.D	Matice sloupcových vektorů, kde každý vektor obsahuje příznaky segmentu.
P.names	Jmenný seznam příznaků
P.class	Pokud jsou jednotlivé segmenty oklasifikovány, pak matice obsahuje třídu každého segmentu.

5.2 Úprava adaptivních segmentů

Funkce `redukce.m` projde matici sloupcových vektorů, kde každý vektor reprezentuje jeden konstantní segment napříč všemi kanály, a odstraní po dvou duplicitní vektory.

Tabulka 2 Funkce pro odstranění duplicitních vektorů `redukce(X)`.

`[Y,a,b] = reduce(X)`	
X	Matice dat, sloupce odpovídají segmentům, řádky příznakům.
Y	Redukovaná matice vektorů z X.
a	Vektor indexů, tak že Y(a) = X.
b	Vektor indexů, tak že X(b) = Y.

Příklad:

$$\mathbf{X} = \begin{bmatrix} 3 & 3 & 3 & 2 & 2 & 3 \\ 2 & 2 & 2 & 1 & 2 & 2 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 3 & 2 & 2 & 3 \\ 2 & 1 & 2 & 2 \end{bmatrix}$$

$$\mathbf{a} = \begin{bmatrix} 1 & 1 & 1 & 2 & 3 & 4 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 1 & 4 & 5 & 6 \end{bmatrix}$$

5.3 Redukce prostoru příznaků

PCA je vypočteno pomocí vlastních čísel a vektorů. Funkce `pca_basis.m` má jako první argument matici sloupcových vektorů příznaků $\mathbf{X} \in \mathbb{R}^{m \times n}$. V případě, že $m < n$ počítá se standardně kovarianční matice $\mathbf{X}\mathbf{X}^T$ a dále vlastní čísla a vektory jako je popsáno v

Výpočet pomocí vlastních čísel a vektorů. Výsledkem je matice Y, která má seřazené hlavní komponenty.

Pokud $m > n$, lze využít implementační trik a počítat menší matici $\mathbf{X}^T\mathbf{X}$ a to následujícím způsobem.

Nechť

$$\mathbf{S} = \mathbf{X}\mathbf{X}^T \qquad \mathbf{T} = \mathbf{X}^T\mathbf{X},$$

pak platí

$$\mathbf{S}\mathbf{u} = \lambda\mathbf{u}$$

$$\mathbf{T}\mathbf{v} = (\mathbf{X}^T\mathbf{X})\mathbf{v} = \lambda\mathbf{v}$$

$$\mathbf{X}(\mathbf{X}^T\mathbf{X})\mathbf{v} = \mathbf{X}(\lambda\mathbf{v}) = \lambda\mathbf{X}\mathbf{v}$$

$$\mathbf{S}(\mathbf{X}\mathbf{v}) = \lambda\mathbf{X}\mathbf{v}.$$

Stačí tedy spočítat vlastní čísla a vektory menší matice $\mathbf{T}$ a výsledné vlastní vektory vynásobit s maticí $\mathbf{X}$.

$$\mathbf{Y} = \mathbf{X}\mathbf{V},$$

kde $\mathbf{V}$ je matice seřazených vlastních vektorů ve sloupcích [9].

Funkce `compact_reprezentation.m` přijímá tři argumenty. Matici dat, které se budou redukovat, matici hlavních komponent, tedy projekční matici, která data promítne do redukovaného prostoru a velikost dimenze.

Tabulka 3 Funkce pro výpočet báze PCA `pca_basis`.

`[Y, lambdas] = pca_basis(X)`

X	Matice dat, sloupce odpovídají segmentům, řádky příznakům.
Y	Seřazené vlastní vektory. Tvoří bázi celého prostoru.
lambdas	Vlastní čísla matice kovariance, odpovídají rozptylům komponent PCA.

Tabulka 4 Funkce pro zobrazení dat do dimenze m `compact_reprezentation`.

`[w] = compact_reprezentation(X, Y, m)`

X	Matice dat k redukci, sloupce odpovídají segmentům, řádky příznakům.
Y	Seřazené vlastní vektory.
m	Velikost výsledného prostoru, do kterého se bude redukovat.
w	Redukovaná data z matice X.

5.4 Učení a klasifikace

Učení a klasifikace je rozdělena na tři části. První část počáteční stavba modelu je počítána vždy a jsou v ní vybrány segmenty, na kterých se model naučí jako první. Následuje učení a klasifikace zbytku záznamu a to ve dvou verzích. Bez využití VD a s jejich využitím.

5.4.1 Počáteční stavba modelu

Pro výběr prvních vzorků ke klasifikaci bylo použito nehierarchické shlukování k-means ve funkci `cluster.m`. Vstupním argument je matice dat X a sloupcové vektory jsou rozděleny do k shluků. Volbu k lze nastavit v rozmezí přirozených čísel a pro každé k bude spočítána varianta rozdělení dat. Pro vlastní spočítání shluků je použita funkce `kmeans`. Aby se snížila pravděpodobnost uváznutí v lokálním extrému je funkce spuštěna několikrát a jako výsledek se bere nejnižší hodnota součtu vzdáleností od středů shluků. Celý tento cyklus je opakován pro proměnný počet shluků. Výsledek je porovnán pomocí průměrů hodnot silhuette grafu 3.1.1 a je zvoleno k podle nejvyšší hodnoty. Z každého shluku je zvolen vektor nejblíže středu.

Tabulka 5 Funkce nalezení středových segmentů shluků `cluster`.

`[s] = cluster(X)`	
X	Matice dat, sloupce odpovídají segmentům, řádky příznakům.
s	Segmenty vybrané do počátečního modelu.

5.4.2 Učení a klasifikace signálu bez užití Voronových diagramů

V každém inkrementálním kroku proběhne učení a následně klasifikace celého záznamu.

Učení

Učení zajišťuje funkce `learn.m`. Přijímá dva argumenty v případě, že model ještě nebyl vytvořen. Prvním argumentem je matice sloupcových vektorů, které mají být zařazeny do modelu a druhým argumentem jejich třídy. Model sestává z KD stromu, který je vytvořen funkcí MATLABu `KDTreeSearcher`.

V případě vytvořeného modelu jsou funkcí přijímány tři argumenty. Stávající model, matice sloupcových vektorů ke klasifikaci a jejich třídy. Výsledkem je aktualizovaný

model. KD strom se po každém přeučení sestaví znovu nejen kvůli zjednodušení implementace, ale i protože po přidávání prvků se strom stává nevyvážený a tedy neefektivní. Vyvažování stromu je implementačně i výpočetně náročně, a proto je jednodušší strom přestavět.

Tabulka 6 Funkce pro učení modelu `learn`.

```
[model] = learn(set, labels)
[model] = learn(model, set, labels)
```

model	Klasifikační model.
set	Vektory, které se přidají do modelu.
labels	Třídy segmentů.
model.kd	KD strom naučené sady.
model.labels	Třídy naučené sady.
model.set	Naučená sada.

Klasifikace

Klasifikace záznamu je implementována ve funkci `clasify_m.m`. Přijímá model vytvořený v 5.4.2 a matici sloupcových vektorů jednotlivých segmentů. Pro praktické využití bude funkci posílána jen ta část, která bude nutná k oklasifikovaní, tedy například pouze okno hodnotícího signálu. Výsledkem je vektoru tříd jednotlivých segmentů. Ve funkci lze určit počet nejbližších sousedů.

Tabulka 7 Funkce pro klasifikaci `clasify_m`.

```
[labels] = clasify(model, X)
```

model	Klasifikační model vytvořený funkcí `learn`
X	Matice segmentů k ohodnocení.
labels	Segmenty vybrané do počátečního modelu.

5.4.3 Učení a klasifikace signálu s užitím Voronových diagramů

Název kapitoly má Voronovy diagramy, ale v implementaci se používá duální graf DT. Prostor nejbližšího souseda tvoří VD, ale v klasifikaci se hledají sousední vektory nově přidaného vektoru a to jsou uzly na hranách nového vektoru grafu DT.
Po každém inkrementálním kroku se přidávají do grafu DT nové vektory a klasifikují se jen ty, které změna ovlivní.

Učení

Funkce učení s Voronovými diagramy `learn_voronoi.m` funguje pouze pro prostor příznaků dimenze 2 a 3. V případě vytváření nového modelu funkce přijímá tři argumenty počáteční množinu segmentů, jim přiřazené třídy a navíc oproti modelu bez VD ještě indexy segmentů v celém signálu. V případě aktualizace modelu přijímá argumenty čtyři, kde čtvrtý je model.

Tabulka 8 Funkce pro učení pomocí VD `learn_voronoi`.

```
[model, changes] = learn_voronoi(set, labels, id)
[model, changes] = learn_voronoi(model, set, labels, id)
```

model	Klasifikační model.
set	Vektory, které se přidají do modelu.
labels	Třídy segmentů.
id	Indexy segmentů v celém signálu
model.kd	KD strom naučené sady.
model.labels	Třídy naučené sady.
model.set	Naučená sada.
changes	Indexy segmentů v signálu, které bude potřeba znovu ohodnotit.

Klasifikace

Funkce `clasify_voronoi` oklasifikuje zadané segmenty a vrátí každému segmentu třídu a nejbližšího souseda.

Tabulka 9 Funkce pro klasifikaci pomocí VD `classify_voronoi`.

```
[labels, neighbors] = clasify_voronoi(model, X)
```

model	Klasifikační model vytvořený funkcí `learn`
X	Matice segmentů k ohodnocení.
labels	Segmenty vybrané do počátečního modelu.
neighbors	Nejbližší sousedé testovacích segmentů.

6 Experiment na EEG záznamech

Byly vyzkoušeny a zhodnoceny výsledky výše popsaných metod. Testováno bylo na novorozeneckých záznamech a artefaktech, segmentovaných adaptivně a osmihodinových datech spánku dospělých, které byly segmentovány konstantně po 30s. Z důvodu nedostatku ohodnocených dlouhodobých záznamů byly v 6.6 spojeny všechny novorozenecké záznamy do jednoho, aby vznikla objemná data. Pro experimenty, kde se posuzuje přesnost klasifikátoru, stačí i kratší data.
Cílem experimentu bylo zredukovat dimenzi prostoru příznaků jak z důvodů vizualizace, abychom o datech získali nějaký přehled, tak hlavně z důvodu výpočtů, neboť časová i paměťová náročnost algoritmů k-means i KNN roste s dimenzí prostoru. Nehledě na Voronovy diagramy, které jsou zde počítány jen pro dvou a třírozměrný prostor.
Pro tento experiment byla nad signály provedena celková datová analýza včetně segmentace v PSGLab toolboxu a veškeré vstupní proměnné implementovaných algoritmů byly již vypočítané příznaky ze segmentů uložené v souboru `arff`.
Součástí experimentu bylo vyzkoušet měření EEG záznamu. Dále v hodnocení ale použit nebyl, protože se nabízel signál se stejnými měřenými artefakty a lepším ohodnocením.
Experiment proběhl v prostředí MATLAB. Extrakce příznaků byla provedena v PSGLab toolboxu [1].

6.1 Použitá data

Použita byla novorozenecká data segmentovaná adaptivně, kde byl hodnocen klidný spánek (QS) a aktivní spánek (AS). Též adaptivně segmentován byl záznam artefaktů s hodnocením normální aktivity, pohybového artefaktu a mrkání. Poslední sadou byly konstantně segmentované spánkové záznamy dospělých se čtyřmi třídami - bdělost, NREM1+NREM2, NREM3+NREM4 a REM.

Tabulka 10 Použitá reálná data a jejich vlastnosti.

Název	Počet tříd	Počet záznamů	Segmentace	Segmentů	Délka
novorozenci-id00[01-20]	2	20	Adaptivní	624-701	cca 11m
artefakty-id0001	3	1	Adaptivní	469	47s
dospeli-30s-id000[1-4]	4	4	Konstantní	941-1003	cca 8h

Tabulka 11 Četnost tříd skupin dat

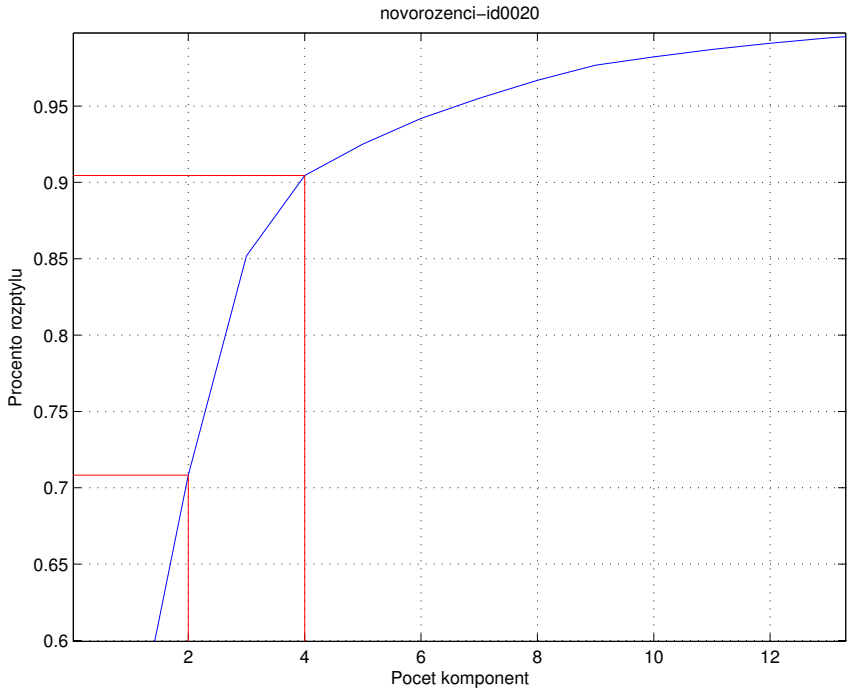
Název	Počet tříd	Četnost tříd
novorozenci-id00[01-20]	2	cca 50% AS, 50% QS
artefakty-id0001	3	63% normální aktivita, 29% pohybový artefakt, 8% mrknutí
dospeli-30s-id000[1-4]	4	21% bdělost, 29% NREM1+NREM2, 31% NREM3+NREM4, 19% REM

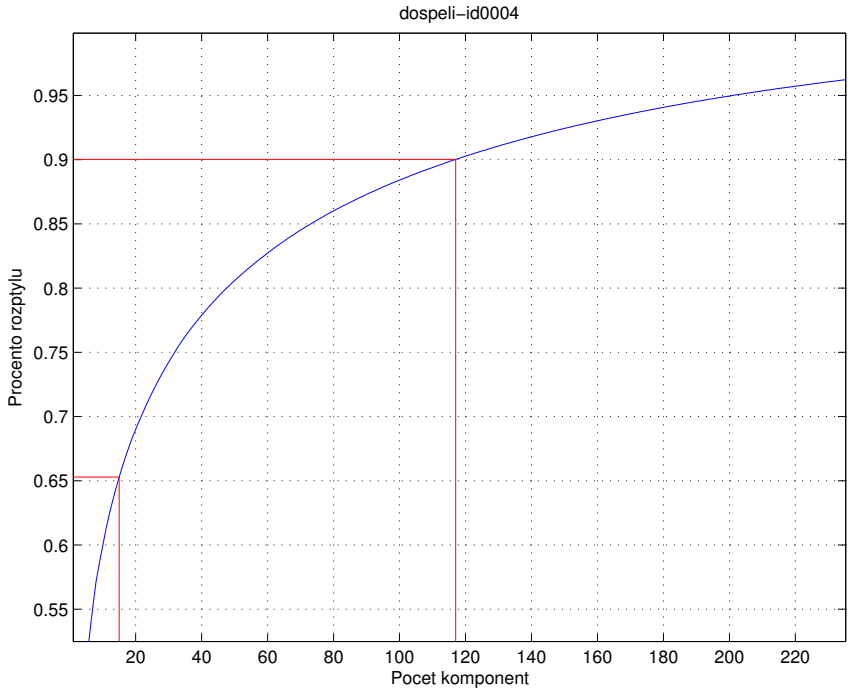
6.2 Redukce dimenze příznaků

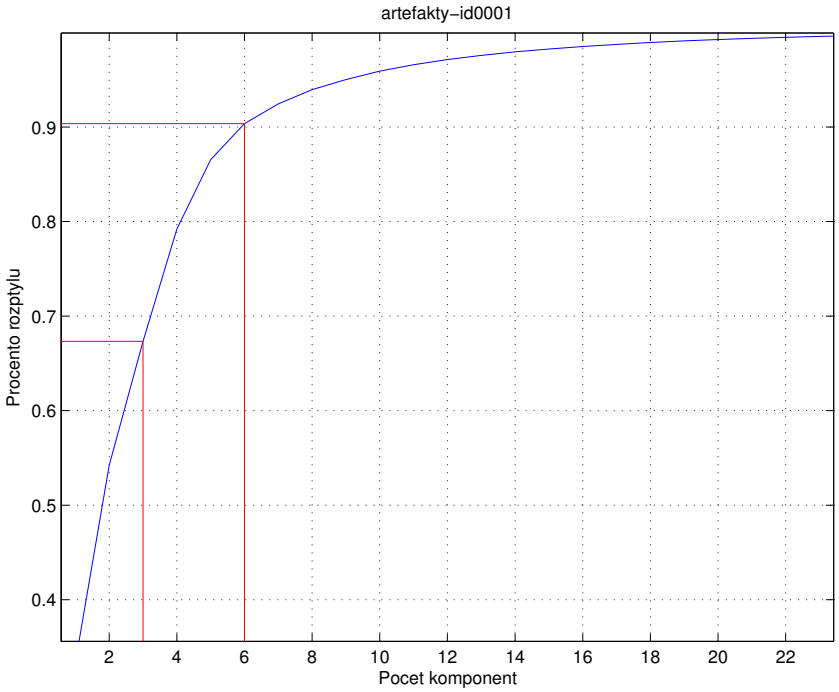
Hlavním cílem redukce dimenze prostoru je urychlit shlukování a klasifikaci pomocí k-means a KNN. Pro vizualizaci byly data zobrazeny ve dvourozměrném prostoru. Metodou pro redukci je analýza hlavních komponent popsaná v 2.5.

6.2.1 Rozptyl komponent PCA

Výpočet rozptylu je popsán v 2.5.2. Na Obr. 10 je vykreslena kumulativní suma rozptylů podle komponent PCA. Počet komponent je zvolen v rozmezí 65% a 90%. U všech dat byl veliký růst funkce a rychlá konvergence k 1, což značí velkou korelaci dat a k dobrému popisu záznamu stačila mnohem menší dimenze příznaků.
Pro popsání 90% rozptylu novorozeneckých dat stačily použít 4 komponenty z celkového počtu 2087. V experimentech klasifikace budou použity 3 komponenty. Pro popis spánku dospělých je potřeba použít komponent více. 90% rozptylu se u testovaných dat pohybovala mezi 100 a 120. Bude použito 100 komponent. Artefaktům vychází hranice 90% na 6 komponent.


novorozenci−id0020
Procento rozptylu
Pocet komponent
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
2
4
6
8
10
12


dospeli−id0004
Procento rozptylu
Pocet komponent
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
20
40
60
80
100
120
140
160
180
200
220



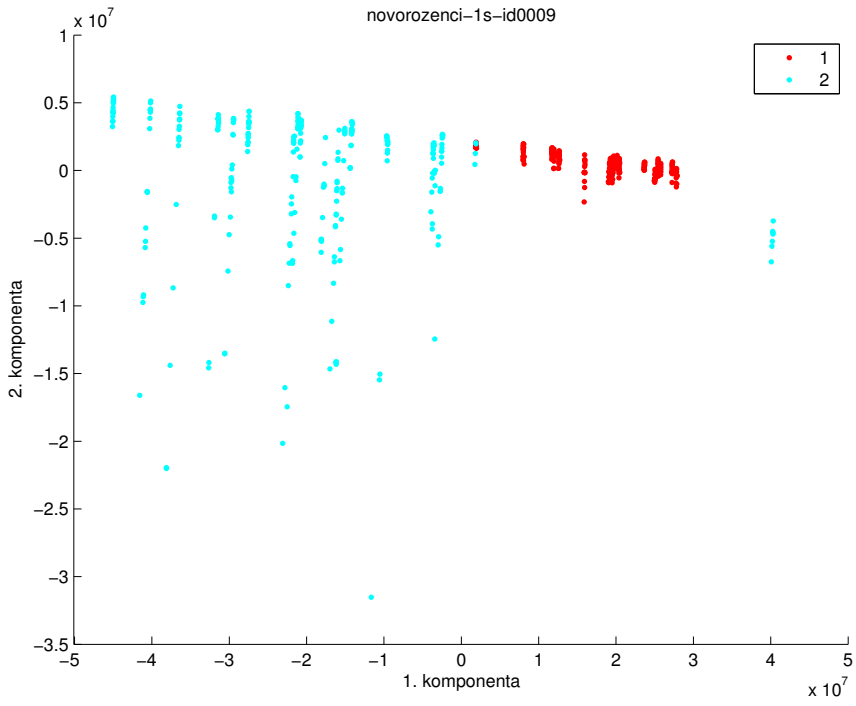
Obrázek 10 Závislost celkového rozptylu dat na počtu komponent PCA. Červené hranice ukazují 65% a 90% rozptylu.

6.2.2 Vykreslení příznaků

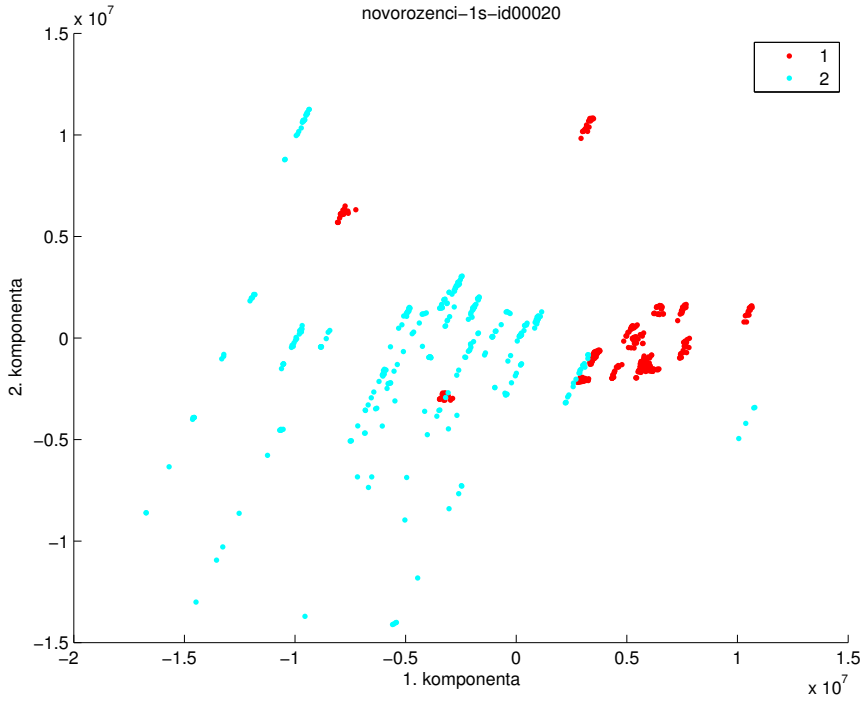
U každého vybraného záznamu byly data zobrazeny ve dvourozměrném prostoru v Obr. 11. Novorozenecké záznamy vytváří už ve dvou dimenzích viditelné shluky tříd AS a QS. Lze proto očekávat, že tato data budou klasifikována velmi přesně.

Záznamy spánku dospělých mají méně separované třídy a rozdíl jejich rozložení mezi různými záznamy je velký. Nejviditelnější třída je NREM3+NREM4 v záznamu *dospeli-30s-id0004*, který je zobrazen světle modrou barvou. Jsou však zobrazeny pouze dvě komponenty (zvoleno bylo 100), takže je to jen velmi hrubý náhled na data.

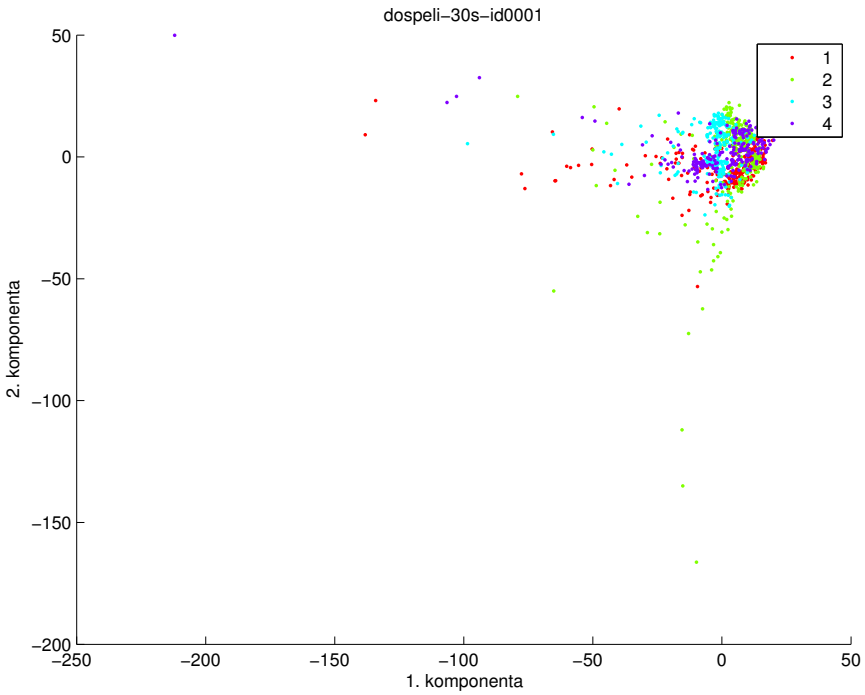
Ani u artefaktů ve dvourozměrném prostoru nejsou třídy viditelně rozděleny.



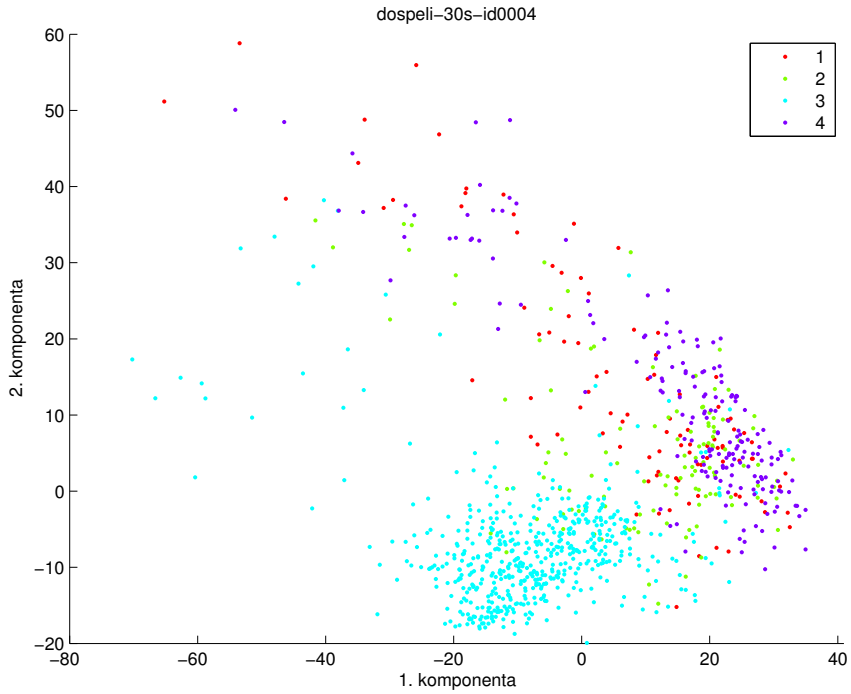
a) Třídy: 1 = QS, 2 = AS



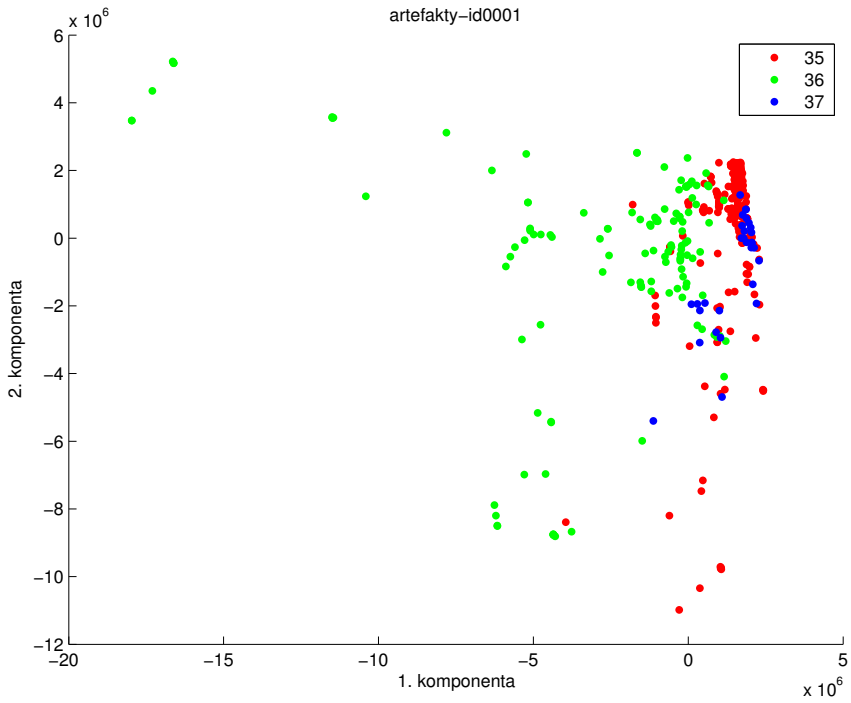
b) Třídy: 1 = QS, 2 = AS



c) Třídy: 1 = bdělost, 2 = NREM1+NREM2, 3 = NREM3+NREM4, 4 = REM



d) Třídy: 1 = bdělost, 2 = NREM1+NREM2, 3 = NREM3+NREM4, 4 = REM



e) Třídy: 35 = normální aktivita, 36 = pohybový artefakt, 37 = mrknutí

Obrázek 11 Vykreslené dvě komponenty PCA.

6.3 Shlukování K-means

Na testovaná data byl použit algoritmus k-means pro odhad shluků. Použití shlukování má za cíl zvýšit přesnost počátečního modelu a snížit počet testovacích vzorků.

6.3.1 Odhad optimálního počtu shluků

Bylo testováno celkem pět záznamů. Pro každý byl automaticky vypočten optimální počet shluků tak, že byly zvoleny varianty pro 2 až 40 shluků a vybráno bylo číslo s nejvyšším průměrem hodnot silhouette grafu. Tab. 12 ukazuje hodnoty okolo optimální vypočítané. Čísla jsou v rozmezí -1 až 1. Vyšší číslo znamená, že shluky jsou více oddělené.

Na Obr. 11 a Obr. 12 při srovnání tříd metoda shlukování u novorozeneckých záznamů naznačuje větší úspěšnost než u artefaktů a spánku dospělých, kde shluky neodpovídají třídám a pro větší úspěšnost modelu je výhodnější volit větší počet shluků, aby trénovací vzorky byly hustěji rozmístěné. Metoda vykazuje lepší výsledky než náhodný

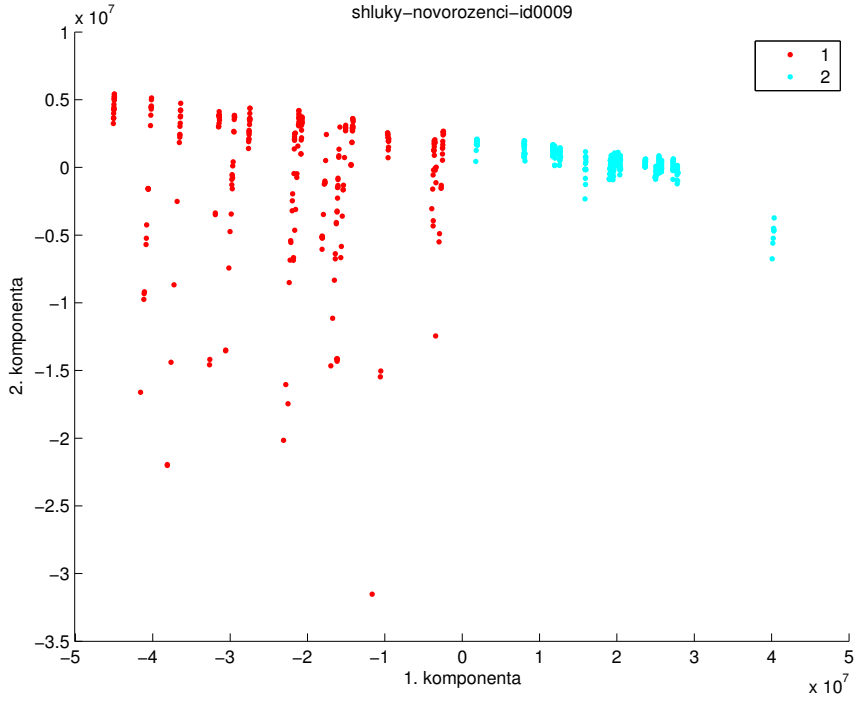
výběr vzorků.

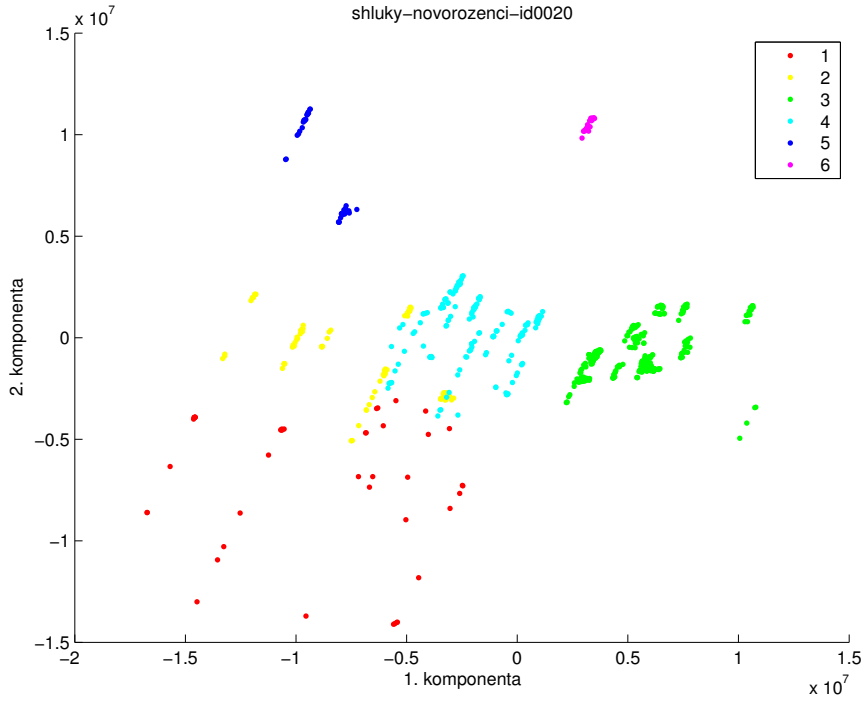
Každému shluku je nalezen prvek nejblíže jeho středu a je přidán do trénovací množiny.

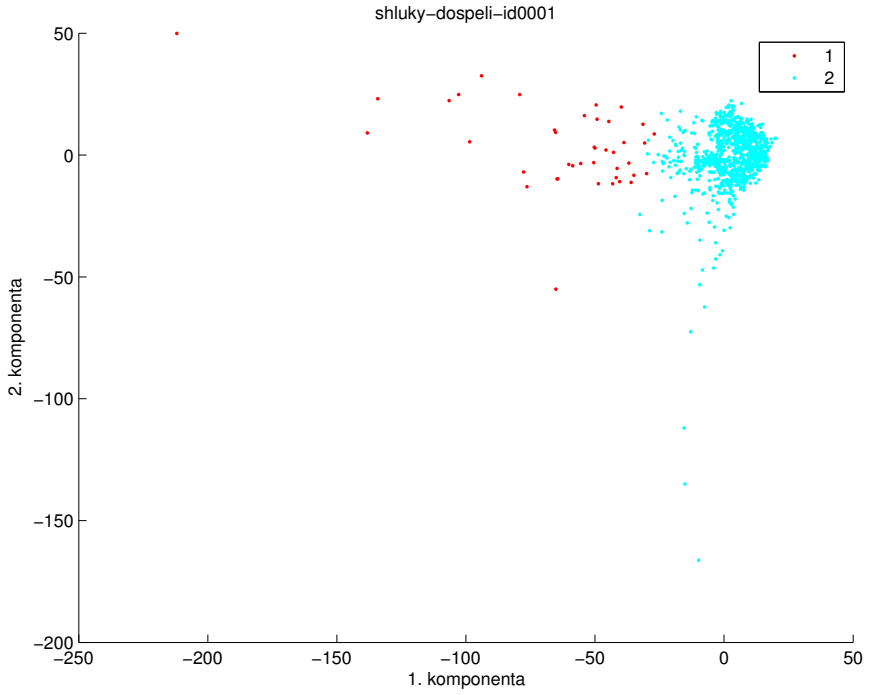
Počet vzorků trénovací sady je roven počtu shluků.

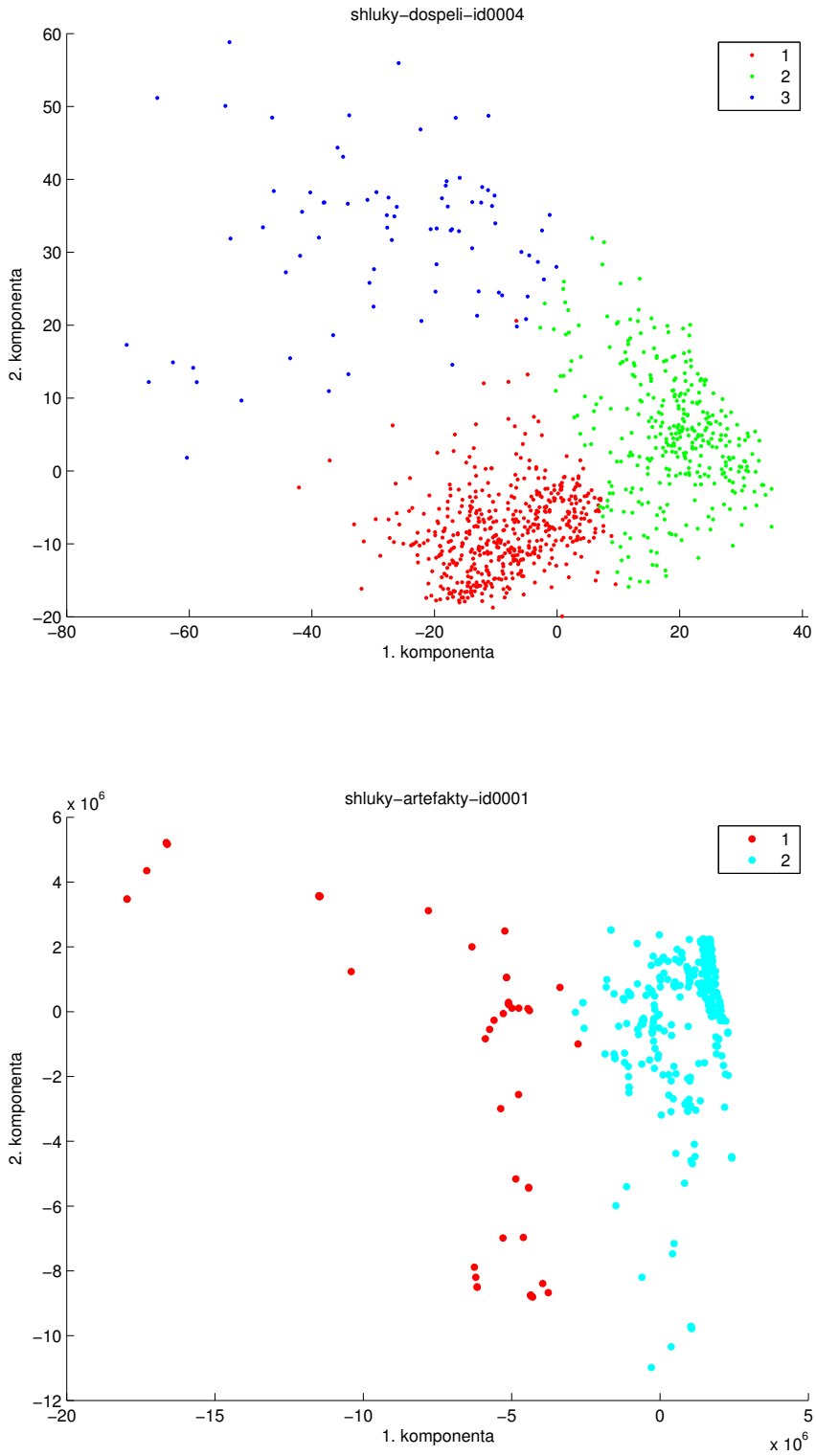
Tabulka 12 Střední hodnoty silhouette grafu při počtech shluků od 2 do 7.

		Počet shluků					
Název	**Počet tříd**	**2**	**3**	**4**	**5**	**6**	**7**
novorozenci-id0009	2	**0.836**	0.776	0.717	0.674	0.642	0.757
novorozenci-id0020	2	0.590	0.634	0.654	0.648	**0.700**	0.642
dospeli-30s-id0001	3	**0.795**	0.178	0.173	0.225	0.208	0.227
dospeli-30s-id0004	3	0.298	**0.367**	0.329	0.187	0.184	0.197
artefakty-id0001	4	**0.855**	0.680	0.587	0.616	0.631	0.697




shluky−novorozenci−id0020
x 10^7
1.5
1
0.5
0
−0.5
−1
−1.5
2. komponenta
−2
−1.5
−1
−0.5
0
0.5
1
1.5
1. komponenta
x 10^7
1
2
3
4
5
6


shluky−dospeli−id0001
50
0
−50
−100
−150
−200
2. komponenta
−250
−200
−150
−100
−50
0
50
1. komponenta
1
2

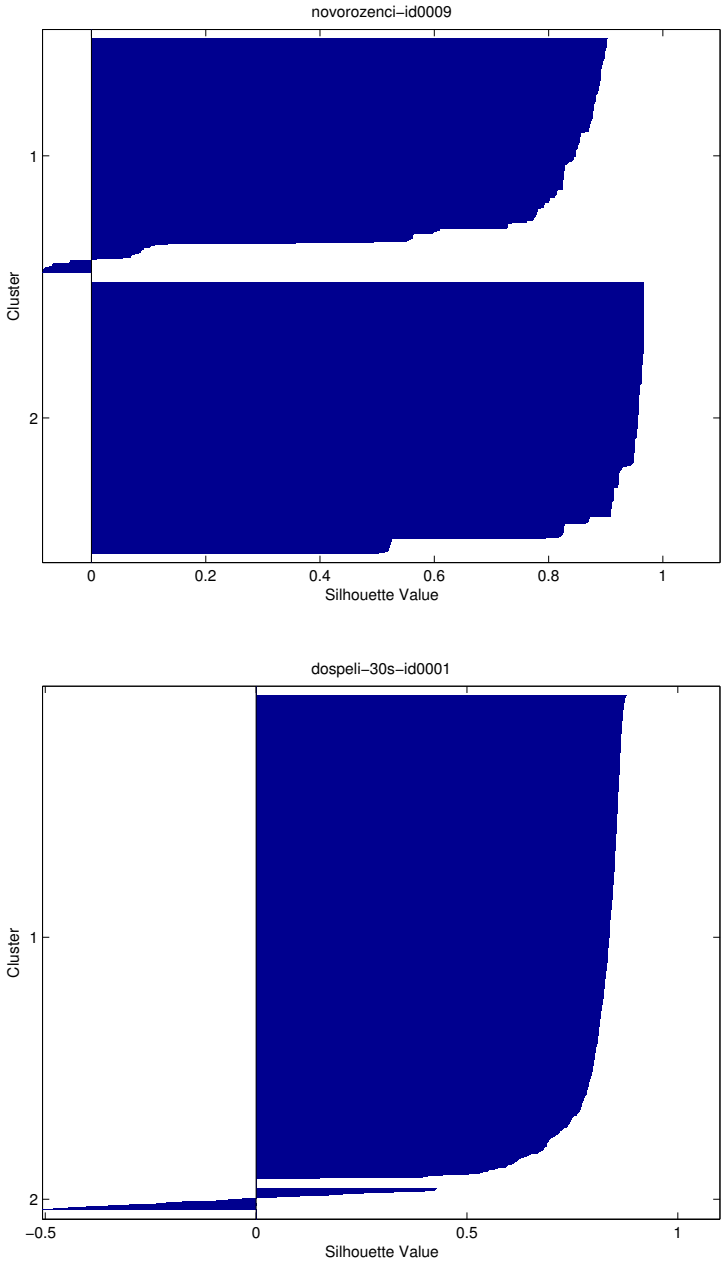


Obrázek 12 Vykreslené první dvě souřadnice vypočtených shluků metodou k-means s automatickým výběrem počtu shluků.

Vykresleny byly jen první dvě souřadnice, jedná se tedy jen přibližné zobrazení dat. Nicméně pro ilustraci výsledných shluků je to postačující.

Záznamy novorozenců byly téměř správně rozděleny do shluků. V případě *novorozenci-id0009* lze očekávat téměř 100% úspěšnost klasifikátoru se dvěma naučenými vzorky a v *novorozenci-id0020* vzniklo šest shluků, které až na pár vzorků oklasifikují celou sadu také správně.

V datech spánku dospělých bude v počátečním modelu správně oklasifikována pouze třída NREM3+NREM4. Pro zbylé třídy nebude klasifikátor fungovat příliš dobře.



Obrázek 13 Rozdíl v silhouette grafech spánku novorozenců a dospělých.

Obr. 13 ukazuje, že v novorozeneckých záznamech jsou shluky dobře separované.

Většina vzorků má hodnotu blízko 1 a jen pár vzorků má záporné hodnoty. Na rozdíl u spánku dospělých, kde je velká nejistota v druhém shluku.

6.4 Přesnost klasifikace počátečního modelu

Počátečním modelem byl oklasifikován celý záznam. Klasifikace probíhala metodou nejbližšího souseda. Zkoumán byl pouze náhodný výběr vzorků a shlukování. Postupné vybírání by nemělo velký smysl u novorozeneckých záznamů, kde první polovina záznamu má třídu QS a druhá AS. Model by tedy neměl o druhé třídě do poloviny záznamu žádné informace.

6.4.1 Klasifikace s náhodným výběrem vzorků

Náhodně byl vybrán určitý počet segmentů do počátečního modelu a zbytek záznamu byl ohodnocen. Celý tento proces byl zopakován 100-krát a výsledek klasifikace zprůměrován v Tab. 13.
U novorozeneckých dat lze pozorovat velkou úspěšnost už při 1% naučených dat, což odpovídá 6 segmentům. Segmentů jedné třídy je přibližně stejně jako druhé třídy. Pravděpodobnost, že v šesti vzorcích bude alespoň jeden zástupce z obou tříd je asi $0,92$.
Náhodné vybírání vzorků je relativně spolehlivou a hlavně rychlou metodou pro počáteční model. U ostatních záznamů úspěšnost kolísá a u spánku dospělých se velmi liší. Například v *dospeli-30s-id0002* model není spolehlivý ani při naučení nad 50% záznamu.

6.4.2 Klasifikace s využitím shluků

Algoritmus k-means byl spouštěn 20-krát, aby bylo dosaženo co nejlepšího výsledku z důvodu možnosti uváznutí v lokálním minimu. Nejdříve se počítala klasifikace s optimálním odhadem počtu shluků 6.3.1. Testováno bylo rozmezí mezi 2 až 40 shluky. Ruční výběr byl stanoven na 4, 10, 20 a 40. Automatický výběr vycházel mezi 2 a 8 shluky pro novorozenecká data a 2 až 5 pro spánek dospělých.
Pro automaticky stanovený počet shluků vychází 75,3% úspěšnost klasifikace v datech novorozenců. Pro počet shluků stanovený na šest na stejném záznamu vychází průměrná úspěšnost 84%. Náhodný výběr se šesti vzorky má úspěšnost 79%. U dat spánku dospělých je průměr automatického výběru 47%. Pro srovnání ruční výběr shluků s

počtem 10 má úspěšnost 52%, náhodný výběr s 9 až 10 vzorky má 49%. Artefakty se 4 shluky 68% a náhodným výběrem se čtyřmi vzorky 70,3%, avšak pro 88% úspěšnost stačí pouze 10 vzorků ze shlukování. Náhodný výběr má takovou úspěšnost se 45 vzorky. Na testovaných datech nelze jednoznačně určit, zda se metoda shlukování vyplatí. Na novorozeneckých datech automaticky vypočtený počet shluků dává dobré výsledky. U ostatních záznamů nejsou shluky tak viditelné a pro lepší výsledky by byly nutné podrobnější analýzy již při výběru příznaků. V dalších testech se bude počítat s metodou shlukování a automatickým výběrem shluků.

Tabulka 13 Ukázka přesnosti klasifikátoru s různou délkou naučeného signálu. U každého procentuálního vyjádření přesnosti je uvedeno v závorce, kolik vzorků bylo v trénovací sadě.

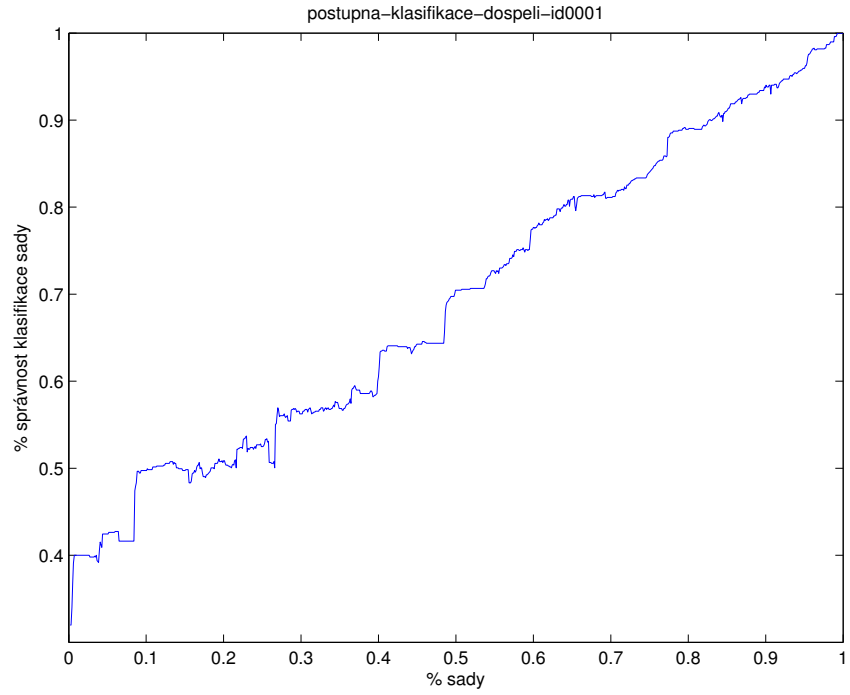
	% naučeného signálu				
Data	**1**	**5**	**10**	**25**	**50**
novorozenci-id0001	74.6 (6)	88.2 (33)	93.6 (63)	98.4 (148)	99.2 (271)
novorozenci-id0002	80.8 (6)	91.7 (32)	95.6 (65)	98.5 (145)	99.3 (259)
novorozenci-id0003	91.7 (6)	95.7 (34)	98.1 (64)	99.5 (156)	99.6 (270)
novorozenci-id0004	65.7 (6)	89.1 (32)	95.2 (62)	98.7 (149)	99.5 (260)
novorozenci-id0005	89.5 (6)	96.5 (33)	98.2 (61)	99.2 (142)	99.4 (266)
novorozenci-id0006	81.5 (6)	89.6 (34)	93.7 (68)	96.8 (154)	97.9 (270)
novorozenci-id0007	74.6 (6)	85.0 (31)	90.6 (60)	95.2 (138)	96.5 (251)
novorozenci-id0008	59.6 (6)	78.0 (33)	85.9 (64)	93.1 (142)	95.4 (269)
novorozenci-id0009	92,0 (6)	96,8 (34)	98,0 (65)	99,0 (154)	99,1 (272)
novorozenci-id0020	80,5 (6)	93,4 (34)	95,9 (62)	98,4 (153)	99,2 (262)
dospeli-30s-id0001	44,7 (9)	61,7 (47)	66,9 (91)	72,6 (218)	76,9 (406)
dospeli-30s-id0002	37,5 (9)	41,3 (47)	42,5 (94)	45,5 (216)	48,2 (387)
dospeli-30s-id0003	51.0 (10)	54.6 (50)	55.5 (94)	56.7 (221)	56.5 (396)
dospeli-30s-id0004	65,0 (9)	73,6 (47)	76,7 (92)	81,1 (213)	85,2 (365)
artefakty-id0001	70,3 (4)	85,2 (23)	88,0 (45)	90,0 (107)	91,0 (181)

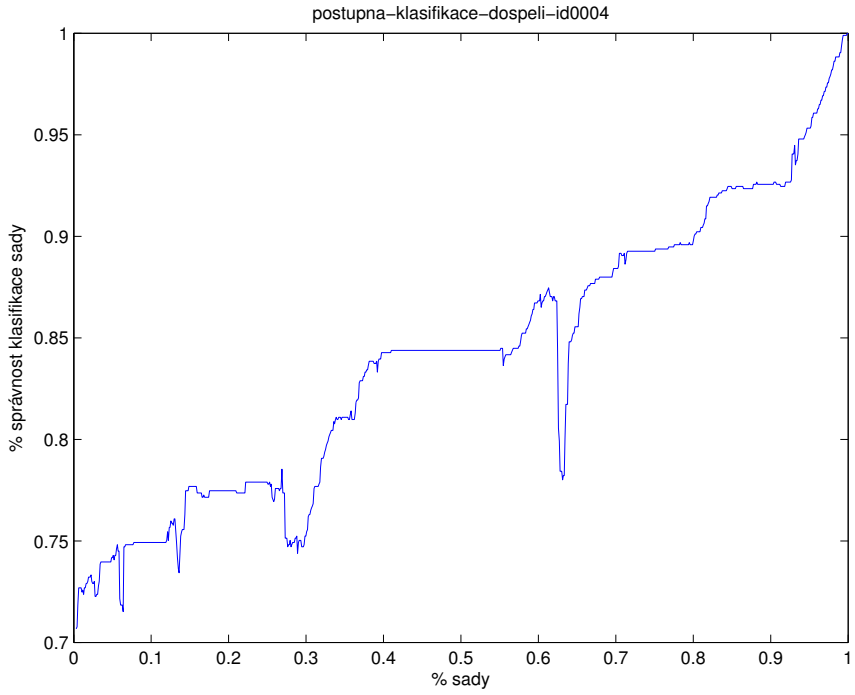
Tabulka 14 Přesnost klasifikace počáteční sadou s využitím různého počtu shluků. První sloupec klasifikace odpovídá automaticky vypočítané hodnotě počtu shluků metodou k-means.

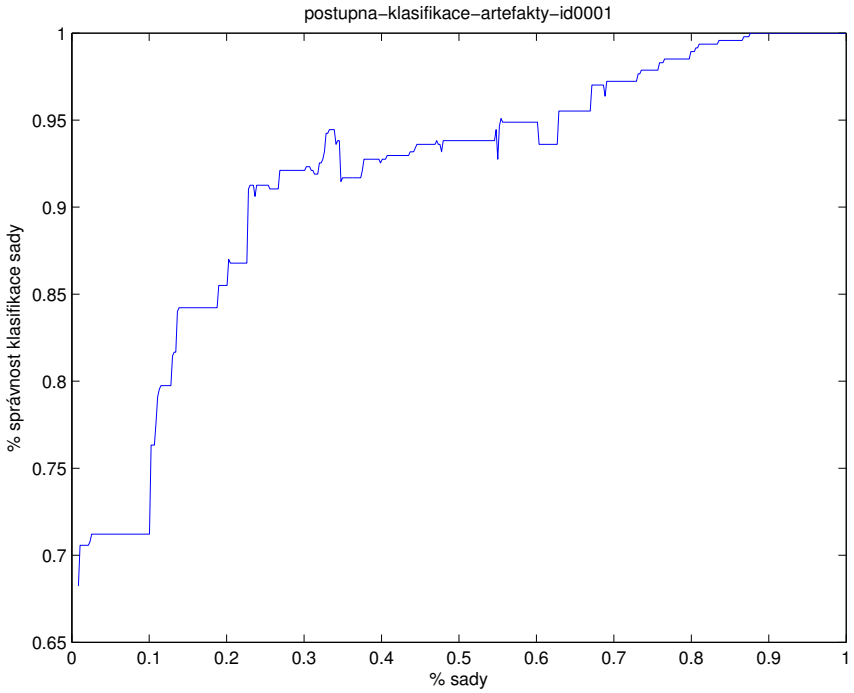
Data	Auto	4	6	10	20	40
novorozenci-id0001	82,0 (8)	74,1	79,5	84,8	81,8	97,5
novorozenci-id0002	84,1 (5)	83,2	90,8	79,3	88,2	97,3
novorozenci-id0003	91,8 (3)	94,7	94,7	94,9	91,0	99,5
novorozenci-id0004	51,2 (7)	48,2	51,3	67,3	82,3	99,0
novorozenci-id0005	76,9 (2)	95,2	95,1	95,1	95,0	99,0
novorozenci-id0006	77,5 (2)	85,0	83,1	79,6	91,8	96,5
novorozenci-id0007	49,3 (2)	49,2	82,8	82,9	88,8	92,4
novorozenci-id0008	51,8 (2)	63,9	72,8	60,5	75,5	87,3
novorozenci-id0009	98,0 (2)	92,1	97,7	96,2	99,4	99,4
novorozenci-id0020	90,3 (6)	87,7	92,2	90,3	94,6	98,5
dospeli-30s-id0001	32,0 (2)	42,2	42,6	55,1	68,1	70,0
dospeli-30s-id0002	28,2 (2)	22,4	40,1	30,2	42,1	45,7
dospeli-30s-id0003	59,2 (5)	62,3	61,6	54,4	52.0	55,6
dospeli-30s-id0004	70,6 (3)	68,2	63,5	67,6	73,1	78.5
artefakty-id0001	28,9 (2)	68,0	75,2	88,3	91,0	90,8

6.5 Inkrementální přidávání

Dále bylo testováno postupné přidávání nových trénovacích segmentů do modelu, tedy inkrementální učení. Jako parametry pokusu byly zvoleny 3 komponenty PCA pro novorozenecký spánek, 100 komponent pro spánek dospělých a 6 komponent pro artefakty. Pro počáteční model byla zvolena metoda shlukování s automatickým výběrem shluků. Z učících křivek je vidět, že celkově dochází k zpřesňování modelu, ale i ke zhoršování. K tomu dochází buď v případech, kdy je málo trénovacích vzorků a další vzorek na rozhraní tříd ovlivní celé své okolí, nebo v patologických případech, kdy je ohodnocen šum. Z obrázků je patrné, že na začátku je téměř vždy propad v přesnosti klasifikace a dále se klasifikace postupně zlepšuje.

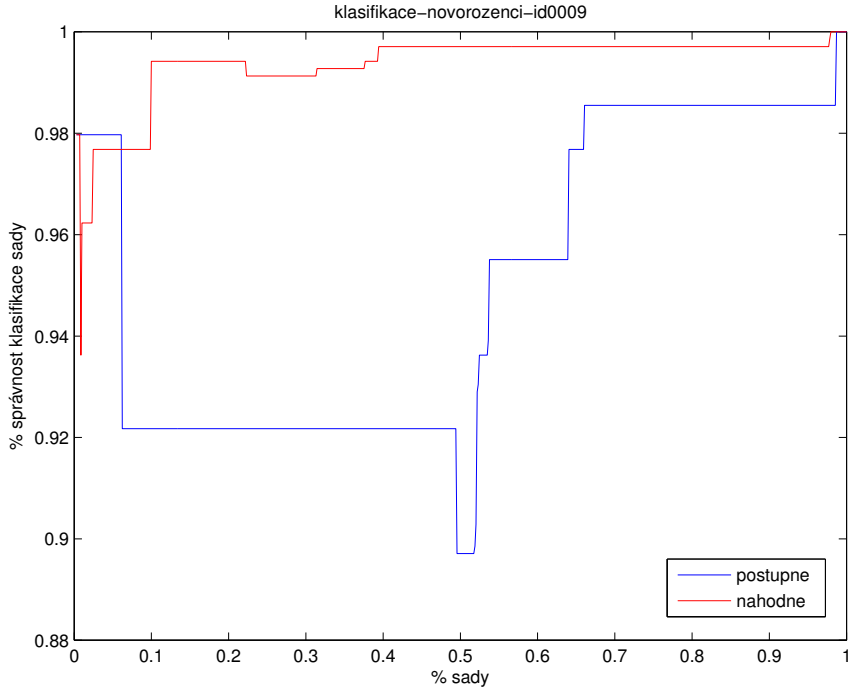

postupna−klasifikace−dospeli−id0001
% správnost klasifikace sady
% sady
1
0.9
0.8
0.7
0.6
0.5
0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

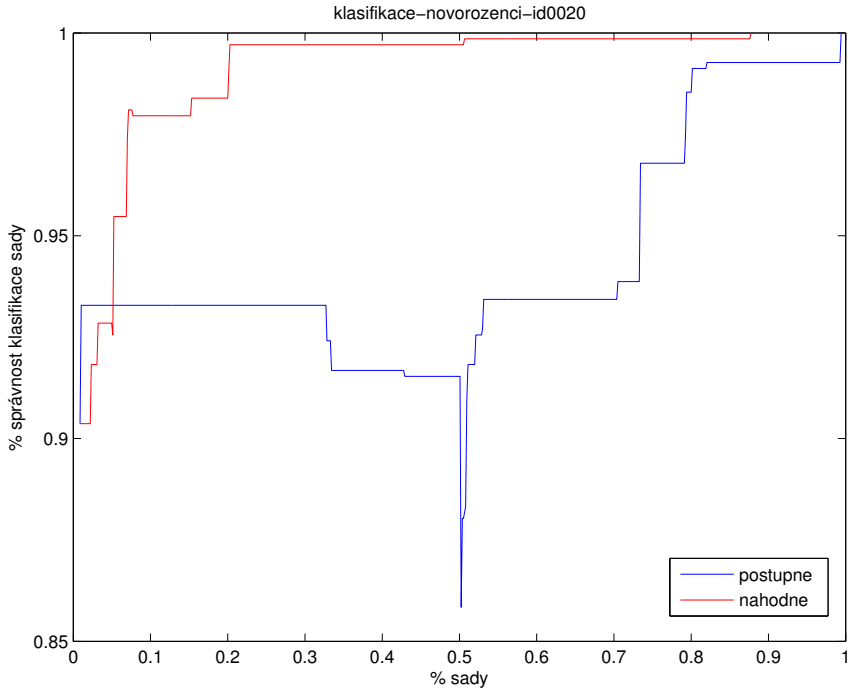

postupna−klasifikace−dospeli−id0004
% správnost klasifikace sady
% sady
1
0.95
0.9
0.85
0.8
0.75
0.7
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Obrázek 14 Zobrazení postupného inkrementálního rozšiřování modelu.

U novorozeneckých záznamů se klasifikátor do 50% sady nezlepšuje. To je způsobeno rozložením tříd, a proto je výhodnější hodnotit záznamy náhodně. Na Obr. 15 je zobrazeno inkrementální učení s postupným i náhodným krokem hodnocení.





Obrázek 15 Srovnání postupného a náhodného inkrementálního rozšiřování modelu u novorozeneckých záznamů.

6.6 Výpočet přes Voronovy diagramy

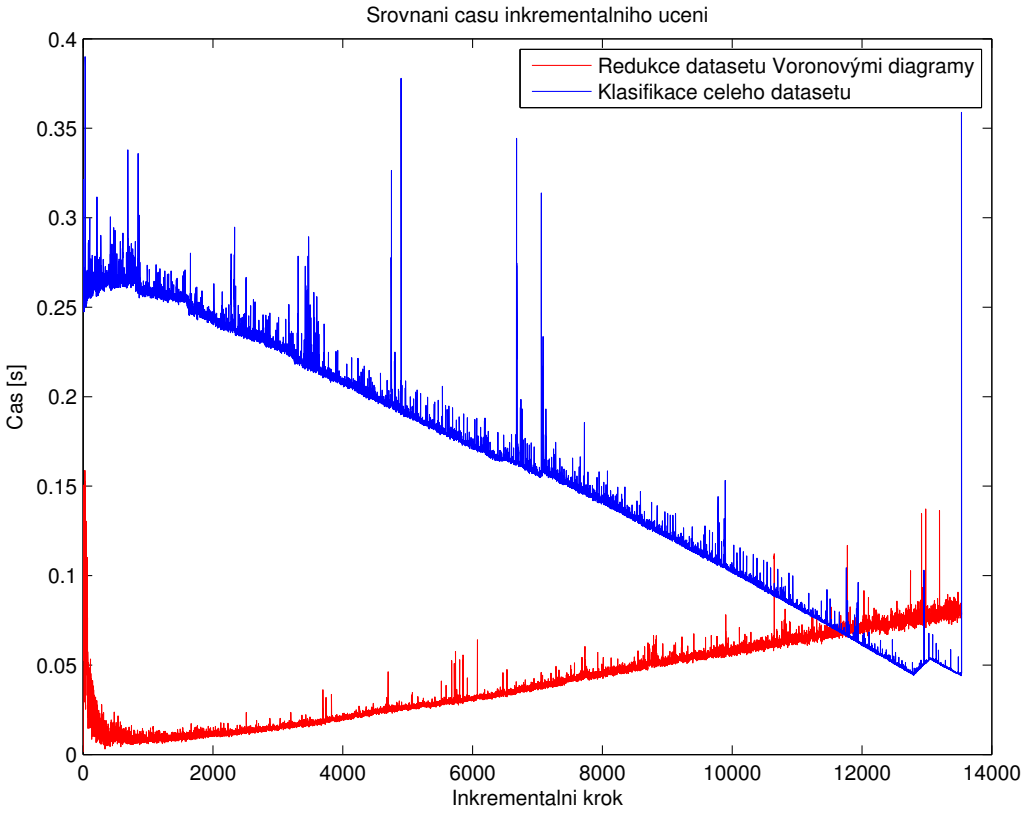
Pro získání delších záznamů byly spojeny všechny novorozenecké záznamy do jednoho. Vznikl tak signál obsahující 13538 segmentů. Záznam byl zredukován do třírozměrného prostoru. Automaticky byl zjišťován nejlepší počet shluků v rozmezí 2 až 10 shluků a opakováním každého výpočtu 10-krát. Čas je zaznamenám ve sloupci *Počáteční model*. Nakonec byl proveden celý postupný inkrementální proces. Časový test tohoto experimentu byl proveden jak Voronovými diagramy, tak i klasickou metodou, kdy se přeučuje celá množina.

Tabulka 15 Časy jednotlivých kroků.

PCA	Počáteční model	Přeučování celé množiny	Přeučování redukované množiny
21s	64s	38min	9min

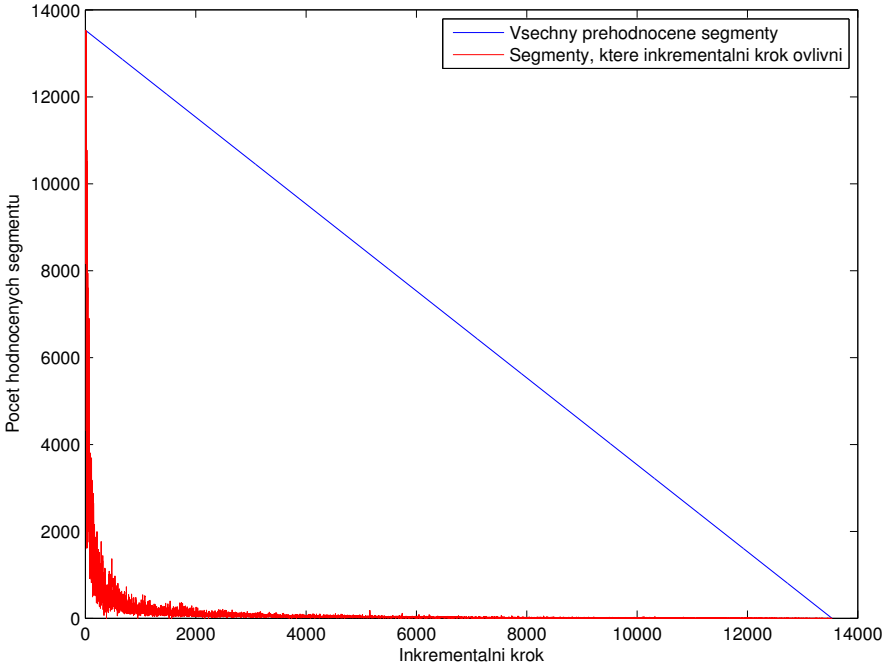
Na Obr. 16 je vidět srovnání časů běhu obou algoritmů. Zatímco čas metody klasifikace celého záznamu se zmenšuje, čas klasifikace pomocí VD roste. Je to způsobeno, že časová náročnost klasifikace celého záznamu klesá a náročnost na přestavění KD stromu se výrazně nezvyšuje. Složitost Voronova diagramu roste jak pro hledání nej-

bližšího souseda, tak pro rozšiřování diagramu.



Obrázek 16 Srovnání času klasifikace celého záznamu a vybraných vzorků pomocí VD.

Na Obr. 17 je znázorněno srovnání počtu testovaných vzorků obou metod. V experimentu testovaných vzorků první metodou ubývá počet testovaných vzorků o jeden za iteraci. Pomocí VD, je množina výrazně zredukována.



Obrázek 17 Srovnání počtu klasifikace celého záznamu a vybraných vzorků pomocí VD.

7 Závěr

Metoda inkrementálního učení EEG záznamu nabízí pomůcku, která ulehčí práci lékařů, ale není tak razantní jako automatická klasifikace.

Připravené metody dokázaly urychlit klasifikaci EEG záznamu. Možnost velké redukce dimenze příznaků snížila metodám k-means a KNN náročnost výpočtu a v případě novorozeneckého spánku použití Voronových diagramů, což znamenalo razantní snížení počtu klasifikovaných segmentů po každém inkrementálním kroku.

Analýzou PCA bylo ukázáno, že data bylo možné redukovat z dimenzí okolo 2000 do dimenzí 2 až 100 bez ztráty přesnosti klasifikace. Na takto redukovaných datech byly vidět rozdíly v rozložení tříd u různých zkoumaných osob, z čehož plyne, že sestavení přesného klasifikátoru, který by fungoval na obecných datech je nereálné.

Metoda shlukování k-means přinesla u záznamu se zřejmými shluky užitek ve formě lepších výsledků klasifikátoru v počátku klasifikace, kdy byl klasifikátor naučen jen na několika segmentech. U ostatních záznamů bylo časově výhodnější vybrat vzorky náhodně.

Klasifikace pomocí VD urychlila čas inkrementálního učení 4-krát a průměrně byla množina testovaných segmentů 62-krát menší. Pro vyšší dimenze se tato metoda použít nedá a jako řešení bylo navrženo klasifikovat jen omezené úseky záznamu, což by bylo využité v grafické nadstavbě.

Implementované metody je možné v budoucnu rozšířit o grafickou nadstavbu (GUI), což umožní pohodlnou klasifikace těchto dat. Dalším vylepšením by mohly být například pokročilejší vizualizace, které by lékařům usnadnily získávání hrubé představy o povaze hodnocených dat a umožnily zobrazit klinicky zajímavé informace.

Literatura

[1] Václav Gerla. *PSGLab | Polysomnographic data processing Matlab toolbox*. 2012. URL: http://bio.felk.cvut.cz/psglab/.

[2] Matej Murgaš. "Incremental Learning in the Task of EEG Signal Classification". bachelor thesis. The Czech Technical University in Prague, 2013.

[3] Tomáš Kaiser. "Analýza epileptických EEG signál·". bachelor thesis. The Czech Technical University in Prague, 2012.

[4] Václav Gerla. "Automated Analysis of Long-Term EEG Signals". doctor thesis. The Czech Technical University in Prague, 2012.

[5] Hojjat Adeli a Samanwoy Ghosh-Dastidar. *Automated EEG-Based Diagnosis of Neurological Disorders: Inventing the Future of Neurology*. 1. vyd. CRC Press, ún. 2010. ISBN: 9781439815311. URL: http://amazon.com/o/ASIN/1439815313/.

[6] Saeid Sanei a J. A. Chambers. *EEG Signal Processing*. 1. vyd. Wiley-Interscience, zář. 2007. ISBN: 9780470025819. URL: http://amazon.com/o/ASIN/0470025816/.

[7] Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002. ISBN: 9780387954424. URL: http://books.google.cz/books?id=%5C_olByCrhjwIC.

[8] Jonathon Shlens. "A tutorial on Principal Component Analysis". In: *Systems Neurobiology Laboratory, Salk Institute for Biological Studies*. 2005.

[9] *courses:ae4b33rpz:labs:12_pca:start [Course Ware]*. 2013. URL: https://cw.felk.cvut.cz/wiki/courses/ae4b33rpz/labs/12_pca/start.

[10] Xin Geng a Kate Smith-Miles. "Incremental Learning". In: *Encyclopedia of Biometrics*. Ed. StanZ. Li a Anil Jain. Springer US, 2009, s. 731–735. ISBN: 978-0-387-73002-8. DOI: 10.1007/978-0-387-73003-5_304. URL: http://dx.doi.org/10.1007/978-0-387-73003-5_304.

[11] Burr Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009.

[12] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), s. 53–65. ISSN: 0377-0427. DOI: http://dx.doi.org/10.1016/0377-0427(87)90125-7. URL: http://www.sciencedirect.com/science/article/pii/0377042787901257.

[13] *k-Means Clustering - MATLAB & Simulink*. 2014. URL: http://www.mathworks.com/help/stats/k-means-clustering.html.

[14] *Silhouette plot*. 2014. URL: http://www.mathworks.com/help/stats/silhouette.html.

[15] *courses:ae4b33rpz:labs:10_k-means:start [Course Ware]*. 2013. URL: https://cw.felk.cvut.cz/wiki/courses/ae4b33rpz/labs/10_k-means/start.

[16] Richard O. Duda, Peter E. Hart a David G. Stork. *Pattern classification*. Wiley New York, 2001.

[17] Tomáš Procházka Michal Houdek Tomáš Svoboda. *Klasifikace podle nejbližších sousedů Nearest Neighbour Classification [k-NN]*. 2001. URL: http://cmp.felk.cvut.cz/cmp/courses/recognition/zapis_prednasky/zapis_01/4/rpz4.pdf.

[18] Alan J. Broder. "Strategies for Efficient Incremental Nearest Neighbor Search". In: *Pattern Recogn.* 23.1-2 (led. 1990), s. 171–178. ISSN: 0031-3203. DOI: 10.1016/0031-3203(90)90057-R. URL: http://dx.doi.org/10.1016/0031-3203(90)90057-R.

[19] Huseyin Akcan. *Kd-Tree Applet*. URL: http://homes.ieu.edu.tr/hakcan/projects/kdtree/kdTree.html.

[20] Andrew Moore. *Animations of KD-tree searches*. URL: http://www.cs.cmu.edu/~awm/animations/kdtree/.

[21] *Nearest neighbor search with kd-trees - ALGLIB*. URL: http://www.alglib.net/other/nearestneighbors.php.

[22] *Voronoi Diagrams and Delaunay Triangulation*. URL: http://www.comp.lancs.ac.uk/~kristof/research/notes/voronoi/dt.gif.

[23] *Attribute-Relation File Format (ARFF)*. URL: `http://www.cs.waikato.ac.nz/ml/weka/arff.html`.