Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

Shlukování textů podle jejich podobnosti pomocí modulu Scikit-learn

Bakalářská práce

Plzeň, 2013

Tomáš Smolík

Prohlášení

Prohlašuji, že jsem předloženou bakalářskou práci vypracoval samostatně na základě konzultací s vedoucí práce a s využitím uvedených pramenů a literatury.

V Plzni dne: 19. srpna 2013 podpis: .....

Poděkování

Rád bych poděkoval Ing. Lucii Skorkovské za odborné vedení práce, čas strávený konzultacemi, řadu přínosných rad, pomoc s hledáním vhodné odborné literatury a v neposlední řadě za mnoho podnětných připomínek. Dále bych rád poděkoval Ing. Luboši Smolíkovi za postřehy týkající se nejen stylistiky, ale také formy práce a Mgr. Evě Smolíkové za práci na gramatických a stylistických korekturách. Také děkuji za podporu celé svojí rodině a svým přátelům.

Anotace

Cílem práce je prozkoumat vybrané algoritmy klasifikace (učení bez učitele) a jejich vhodnost vzhledem k reálnému problému. Tímto problémem je shlukování, respektive dělení novinových článku do skupin v závislosti na jejich tématu. Vybrané algoritmy jsou K-means, analýza hlavních komponent a latentní sémantická analýza. Práce se kromě teoretického úvodu zabývá také experimentální částí, kde jsou vybrané metody otestovány dle určených kritérií.

Klíčová slova: klasifikace, učení bez učitele, shlukování, K-means, analýza hlavních komponent, latentní sémantická analýza

Abstract

The goal is to explore the selected classification algorithms (unsupervised learning) and their suitability for the real problem. This problem is the clustering or separation of newspaper articles into groups depending on their topic. The selected algorithms are the K-means, principal component analysis and latent semantic analysis. The work in addition to theoretical introduction also deals with the experimental part, where some methods are tested according to specific criteria.

Key words: classification, unsupervised learning, clustering, K-means, principal component analysis, latent semantic analysis

Obsah

Kapitola 1

Úvod

1.1 Cíl bakalářské práce

Cílem práce je prozkoumat vybrané algoritmy klasifikace (učení bez učitele) a jejich vhodnost vzhledem k reálnému problému. Vybrané algoritmy jsou K-means, analýza hlavních komponent a latentní sémantická analýza. Tímto problémem je shlukování, respektive dělení novinových článku do skupin v závislosti na jejich tématu. Protože se jedná o typ shlukování, čili způsob učení bez učitele, rozumíme tématem článku vlastně výskyt slov v těchto dokumentech a jejich relace. Práce se kromě teoretického úvodu zabývá také experimentální částí, kde jsou vybrané metody otestovány dle určených kritérií. Vybraná kritéria jsou Homogeneity, Completeness, V-measure a Adjusted Rand Index. Každý ze třech nezávislých experimentů je zacílen na jiné úskalí problému.

1.2 Motivace

Čím dál tím větší množství nestrukturovaných dat [1] nás doslova vybízí k jejich zpracování za účelem získání nějakých konkrétních výstupů. Bohužel však není možné všechna tato data procházet a zpracovávat, neboť by to znamenalo obrovské množství času a tedy mimo jiné i finančních prostředků. Zaměříme-li se pouze na problém, kterým se zabývá

[1] Čili dat, která nejsou například v tabulkách, databázi, nebo jiné struktuře a není možné automaticky pracovat s jejich významem.

tato práce, tedy třídění novinových článků dle tématu, musíme si uvědomit, že jen v České republice denně vzniká tisíce novinových článků. Jen přečíst tento počet článků by bylo pro jednoho člověka v podstatě nemožné. Chceme-li přesto získat z tohoto souboru nějaké informace, je nutné je zpracovat jej počítačově. Jedním z nabízejících se řešení je právě shluková analýza, která na základě podobnosti článků může pomoci vyhledat jisté vnitřní vzory a analogie nestrukturovaných dat.

1.3 Shluková analýza

Shluková analýza [2] je metoda používaná ke klasifikaci objektů nebo jejich obrazů. Klasifikací je míněno třídění, tj. zařazování do různých tříd či skupin. Potřeba takovéto klasifikace může vyvstat v mnoha oborech od biologie, zoologie, psychiatrie, přes sociologii, archeologii, geologii, geografii a samozřejmě také ve strojírenství a informačních technologiích. Se shlukovou analýzou se se můžeme setkat pod různými jinými názvy, jako je numerická taxonomie v biologii a ekologii, typologie ve společenských vědách nebo učení bez učitele [3] v teorii rozpoznávání obrazů.

Použitím shlukové analýzy zařadíme každý z obrazů do skupin, kde jsou si jednotlivé obrazy podobnější než ty z rozdílných skupin. Rozdělení provádíme na základě podobnosti v předem určených kritériích. Každý objekt je tedy definován řadou znaků, které má význam v dané množině sledovat.

Uveďme si to na následujícím příkladu. Máme skupinu zvířat: pes, netopýr, kosatka (savci), želva, varan, krokodýl (plazi), vrabec, tučňák a pštros (ptáci). Chceme-li aplikovat metody shlukové analýzy na soubor těchto živočichů, musíme si nejdříve stanovit kritéria třídění. Například pokud by kritériem byla existence křídel, v jednom shluku by se ocitl netopýr a vrabec, ale také tučňák a pštros (ostatní živočichové tvoří druhou skupinu). Pokud by však kritériem byla schopnost letu, tvořil by první shluk pouze netopýr a vrabec, všechna ostatní zvířata by tvořila druhý shluk. Zkusme ještě případ, kdy by kritériem pro

[2] též clusterová analýza, z anglického cluster = shluk
[3] unsupervised learning, či learning without a teacher

shlukování bylo, zda se jedná o živočicha žijícího ve vodě, či živočicha žijícího na souši. První shluk zjevně vytvoří pes, netopýr, varan, vrabec a pštros, kteří žijí pouze na souši. Na druhou stranu kosatka žije pouze ve vodě a sama tak vytvoří druh shluk. Živočichové, kteří část svého života tráví ve vodě a část na souši, tedy želva, krokodýl a tučňák, pak vytvoří třetí shluk. Pochopitelně je možné dělicí kritéria i nadále přidávat.

Shlukování je jedním z nejzákladnějších myšlenkových postupů lidské mysli, které umožňuje zvládnout přísun obrovského množství informací každý den. Zpracovávání každé informace jako jednotlivého subjektu by bylo totiž nemožné. Proto lidé obvykle vytvářejí vlastní shluky pro kategorizování vjemů (předmětů, osob, událostí, atd.), každý ze shluků je pak charakterizován sadou obecných znaků, na jejímž základě je identifikován [2].

Například většina lidí má vytvořen shluk pes následujícím způsobem: čtyřnohá šelma s prodlouženou čenichovou partií, výrazné ušní boltce, dychtivé oči s typickým výrazem, třeba i vyplazený jazyk, ostré zuby, pronikavý hlas a zvuk, který lze identifikovat jako zřetelné HAF! Uvidí-li pak libovolného psa na základě obecných znaků charakterizujících tento shluk jej správně přiřadí.

Kapitola 2

Teoretická část

2.1 Klasifikace

Jednou z podskupin umělé inteligence je strojové učení, čili klasifikace. Tento obor se zabývá vývojem algoritmů, které mají za cíl umožnit změnit vnitřní stav systému tak, aby zefektivnily jeho schopnosti reakce na změny okolního prostředí. To znamená, že se systém učí. Cílem klasifikace je tedy napodobit myšlení podobně, jako to dělá člověk. Počítačový program by měl reagovat na vstupy, se kterými se doposud nesetkal, a to na základě více či méně podobných situací, které se již „naučil“ [10].

Algoritmy klasifikace lze dělit podle požadovaného výsledku a typu vstupu v průběhu učení.

2.1.1 Supervised learning (učení s učitelem)

První skupinou algoritmů je učení s učitelem. Jedná se o algoritmy vytvářející funkci, která generuje mapu vstupů a k nim odpovídajících výstupů. Ty jsou obvykle určeny expertem. Data, jež jsou nutná k vytvoření této funkce nazýváme tréninková data. Optimální scénář umožní programu správně určit umístění vstupů, které nebyly součástí trénovacích dat.

2.1.2 Unsupervised learning (učení bez učitele)

Učení bez učitele jako druh algoritmů se využívá v případech, kdy se snažíme nalézt vzory v nestrukturovaných datech. Obvykle však není možné výsledky vyhodnotit, protože nelze jednoznačně určit odpovídající strukturu v neznámých datech. Jedinou možností je výstup srovnat s odhadem experta. Tato vlastnost silně odděluje kategorie učení s učitelem a bez něj. Přístup unsupervised learning úzce souvisí se statistickým odhadem funkce pravděpodobnosti. Nicméně zahrnuje i řadu dalších technik, které se snaží shrnout klíčové vlastnosti dat. Často využívají také metody pro předzpracování dat (preprocessing). Tímto způsobem klasifikace se zabývá experimentální část práce.

2.1.3 Semi-supervised learning

Semi-supervised learning je kombinací dvou předchozích. Obvykle zahrnuje malé množství trénovacích dat a velké množství dat neoznačených. Bylo zjištěno, že i malá část trénovacích dat může silně přispět ke zlepšení výsledků učení. Tento způsob je vhodný také v případech, kdy by vytvoření kompletních a často velice rozsáhlých trénovacích dat bylo příliš časově nebo ekonomicky náročné.

2.1.4 Learning to learn

Nebo také Multi-task learning [1] je přístup, který se snaží společně se studovaným problémem získat současně i znalosti o souvisejících problémech. To často vede k lepšímu modulu řešení, protože systém může použít shodnost mezi různými úkoly. Cílem MTL je zlepšit výkonnost algoritmů, které využívají klasifikátor pro řadu podobných úkolů. Klasickým příkladem by mohl být spam filtr e-mailové schránky. Mnoho nevyžádané pošty má podobný obsah, ať už se jedná o hoax [2] či reklamní sdělení. Každý rozeznaný, či označený spam (studovaný problém) tedy pomůže řešit budoucí podobné zprávy (související problémy).

[1] Multi-task learning, zkráceně MTL.
[2] Hoax řetězový e-mail, obvykle mystifikace či žert.

2.1.5 Developmental robotics

Vývojová robotika, někdy nazývaná epigenetická robotika, je vědní obor, který se zaměřuje na studium vývojových mechanismů, architektury a omezení, které by umožnily celoživotní a otevřený průběh učení nových dovedností a nových poznatků systému. Očekává se, že podobně jako u dětí je učení kumulativní a postupně složitější a to z důvodu vlastního zkoumání světa v kombinaci se sociální interakcí. Experimentování s podobnými modely a jejich konfrontování s reálnými problémy poskytuje vědcům zpětnou vazbu a možnost tvořit nové hypotézy v teorii lidského a zvířecího rozvoje.

2.2 Definice shlukování

Definice shlukování vede přímo k definici shluku jako takového. Mnohé definice, které vznikly v průběhu let, používají volně definované podmínky a pojmy jako „podobný“, nebo jsou platné jen pro určitý druh shluků. Tato skutečnost ukazuje, jak obtížné je vytvořit obecně přijatelnou definici shluku. Obecně nejpřijímanější definicí je ta, která definuje shluk jako kontinuální oblast n-rozměrného prostoru s velkou hustotou bodů, oddělená od ostatních shluků prostorem s malou hustotou bodů. Tato definice je blízká naší představivosti pro dvou či třídimenzionální prostor, a proto jsou shluky určené touto definicí někdy nazývány „přírodní shluky“.

Nyní si vysvětlíme jednu z definic, která nám umožní lepší představu o tom, co shlukování vlastně je. I když nemusí být univerzální.

Řekněme, že množina X jsou naše data, respektive příznakové vektory sledovaných objektů:

$$X = \{x_1, x_2, x_3, ..., x_n\}$$

Definujeme m-shlukování X do m shluků $C_1, C_2, ..., C_m$, jsou-li splněny tyto tři podmínky:

- $C_i \neq \emptyset; i \in \{1, 2, ..., m\}$
- $\bigcup_{i=1}^{m} C_i = X$
- $C_i \bigcap C_j = \emptyset; i \neq j \wedge i, j \in \{1, 2, ..., m\}$

Navíc si jsou vektory obsažené ve shluku C navzájem „více podobné“ a zároveň si jsou „méně podobné“ s vektory z jiných shluků. Kvantifikace výrazů více a méně podobné však silně závisí na typu shluků. Například pro kompaktní shluky můžeme podobnost zaměnit s euklidovskou vzdáleností. Pro podlouhlé či dokonce sférické shluky však už musíme zavést speciální pravidla stanovující míru podobnosti vektorů [2].

2.3 Postup při shlukové analýze

Shluková analýza obvykle probíhá v několika základních krocích . Jsou to:

- **Výběr příznaků.** Přáznaky musejí být vybrány tak, aby uchovávaly co nejvíce informací vzhledem ke zkoumané oblasti. Naším cílem je minimalizovat nadbytečné informace. Proto je někdy nutná předpříprava dat před samotným shlukováním. Každý z objektů by měl být reprezentován příznakovým vektorem, tedy vektorem vyjadřujícím příslušnost objektu ke každému z pozorovaných příznaků.

- **Míra podobnosti** je míra, která říká, jak podobné či nepodobné jsou si dva příznakové vektory. Je dobré zajistit, aby všechny prvky přispívaly stejnou měrou do výpočtu míry podobnosti a žádný tedy nebyl dominantní. Na tuto skutečnost by se měl brát ohled již při předpřípravě dat. Zde bodě však velmi záleží na použité metodě a shlukovacím kritériu.

- **Shlukovací kritérium.** Volba kritéria záleží na odhadu experta, který na základě obecné znalosti objektů vyjádří shlukovací kritérium pomocí cenové funkce nebo jiného druhu pravidel. Kritérium vyjadřuje rozumné uspořádání v závislosti na typu shluku (například různé tvary shluků mohou být závislé na různých kritériích).

- **Shlukovací algoritmus.** Tento krok zahrnuje vybrání správného algoritmu v závislosti na znalostech z předchozích dvou kroků.

- **Validace výsledků.** Jakmile obdržíme výsledky shlukování, je důležité je ověřit. O to se obvykle stará příslušný test, který popřípadě odhaluje chyby v některém z předchozích kroků.

- **Interpretace výsledků.** V mnoha případech musí expert sloučit výsledky shlukování s dalšími znalostmi, jako jsou výsledky experimentů a analýz, aby získal správné závěry.

V některých případech je dobré do procesu shlukování zavést část zvanou **shlukovací tendence**. To zahrnuje řadu testů, které naznačují, zda naše dostupná data mají strukturu shluků, či nikoliv. Například odhalí, že hledání shluků v náhodném rozložení objektů je zbytečné.

Jak lze z předchozích odstavců předpokládat, změnou sledovaných znaků (příznakového vektoru), míry podobnosti a shlukovacího kritéria vede shluková analýza ke zcela jiným výsledkům. Ani u nejbanálnější úlohy nelze však jasně určit, které řešení je správné, neboť záleží na jeho aplikaci. Představme si, že máme soubor tří objektů, které leží ve vrcholech rovnostranného trojúhelníku. Můžeme je rozdělit do tří samostatných shluků, nebo do dvou shluků (jeden objekt v prvním shluku a dva objekty v dalším), nebo všechny mají náležet jednomu shluku? Všechna řešení jsou platná, ale které shlukování je správné? Pokud si chceme odpovědět na tuto otázku, potřebujeme se rozhodnout, které z nich nejlépe odpovídá naší aplikaci. Proto je konečná odpověď na tuto otázku dána znalostmi experta [2].

2.4 Příznakový vektor, příznaky

Příznakový vektor je n-rozměrný vektor, který si obecně můžeme představit jako bod v prostoru dimenze n. Prvky vektoru (poloha bodu v prostoru) jsou pak dány jednotlivými příznaky, které určují popisovaný objekt.

Příznak zkoumaného objektu může nabývat jakékoliv reálné hodnoty, nebo může být prvkem konečné diskrétní množiny. Má-li tato sada právě dva prvky (například 0 a 1), nazýváme ji binární či dichotomická. Rozlišujeme čtyři základní kategorie funkcí určující znaky [2]:

- **Nominální.** Příkladem je funkce, která popisuje pohlaví jednotlivce, nabývá například hodnot 1 pro samce a 0 pro samice. Je tedy zřejmé, že jakékoliv kvantitativní srovnání mezi těmito hodnotami je nesmyslné.

- **Ordinální.** Příkladem může být například funkce, která hodnotí studenta podle jeho výkonu. Může například nabývat hodnot 1, 2, 3 nebo 4, které odpovídají hodnocení „výborně“, „velmi dobře“, „dobře“ a „nedostatečně“. Je vidět, že tyto hodnoty jsou uspořádané ve smysluplném pořadí, nicméně rozdíl mezi dvěma po sobě jdoucími hodnotami také nemá kvantitativní význam.

- **Závislé na rozdílu (interval scaled).** Příkladem by mohla být funkce měření teploty na dvou vzdálených místech. Řekněme, že v Praze byla naměřena teplota 10 °C, zatímco v Miami teplota 20 °C. Má smysl říci, že rozdíl teplot mezi Prahou a Miami je 10 °C, ale je nesmyslné obecně tvrdit, že Miami je dvakrát teplejší než Praha, protože funkce podílu teplot v Praze a Miami nemá konkrétní význam.

- **Závislé na podílu (ratio scaled).** Příkladem může být například funkce hmotnosti objektu. Váží-li první objekt 200g a druhý 1000g, můžeme o něm obecně říci, že je pětkrát těžší.

2.5 Použití shlukování

Shlukování je důležitý nástroj používaný v nepřeberném množství různých aplikací a lze ho využít k vyřešení nebo alespoň zjednodušení některých obecných problémů [2]:

- **Redukce objemu dat.** V mnoha oblastech je objem dostupných dat tak obrovský, že jejich zpracování je už příliš náročné. Shlukování lze využít k tomu, aby se data roztřídila do skupin. Pak zpracováváme shluky místo jednotlivých dat. Tím vzniká datová komprese.

- **Vytvoření hypotézy.** V tomto případě aplikujeme shlukovou analýzu na data, abychom odvodili hypotézu vzhledem k povaze dat. Takto získaná hypotéza se musí ověřit například použitím jiných vstupních dat.

- **Testování hypotézy.** Shlukovou analýzu lze využít i pro ověření platnosti specifických hypotéz. Například mějme tvrzení: „Velké společnosti investují v zahraničí.“ Jeden ze způsobů jak ověřit tuto hypotézu je použít shlukování na velkou reprezentativní množinu společností. Každá ze společností bude pak reprezentovaná tříprvkovým příznakovým vektorem s indexy značícími velikost společnosti, míru aktivit v zahraničí a schopností úspěšně dokončovat projekty. Pokud se po použití shlukové analýzy vytvoří shluk odpovídající velkým společnostem aktivních v zahraničí bez ohledu na jejich schopnosti, můžeme hypotézu prohlásit za experimentem podpořenou.

- **Odhad na základě skupin.** Opět použijeme shlukovou analýzu na dostupná data. Výsledné shluky jsou charakterizovány pomocí vzorů, podle kterých byly utvořeny. Jak se ukazuje, pro objekt s neznámým vzorem můžeme určit shluk, do kterého s největší pravděpodobností náleží. Například mějme případ, kdy aplikujeme shlukovou analýzu na množinu dat o pacientech trpících stejnou nemocí. Ti jsou rozděleni do shluků na základě jejich reakce na různé metody léčby. Pro nového pacienta pak najdeme analýzou patřičný shluk podle jeho příznakového vektoru a vybereme pravděpodobně nejvhodnější metodu léčby.

Kapitola 3

Použité metody

3.1 Shlukovací metody a metody výběru znaků

3.1.1 K-means

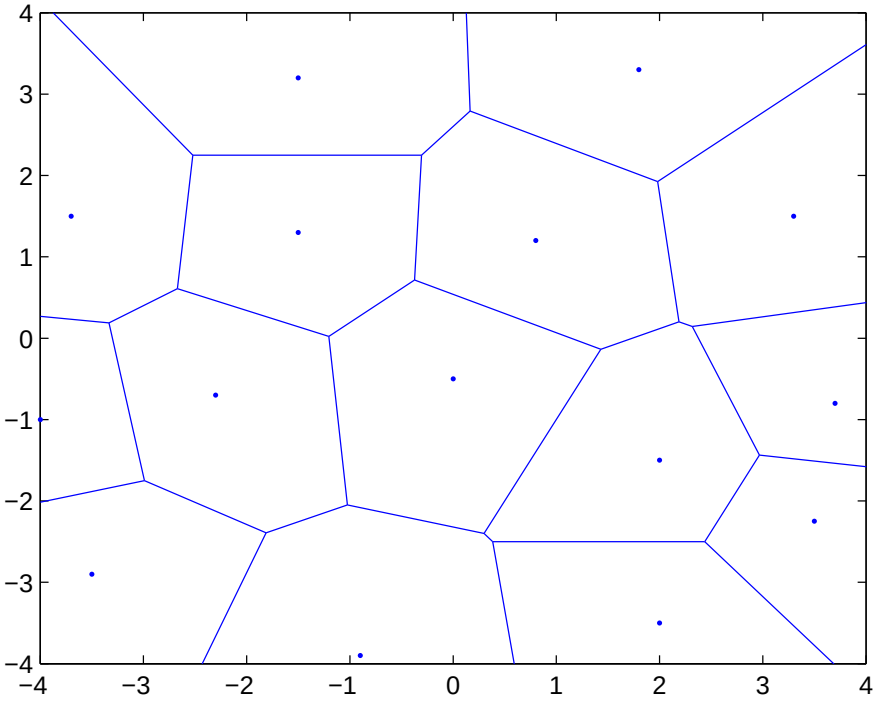
K-means [1] patří mezi nejzákladnější algoritmy shlukovací analýzy. Jeho hlavní myšlenka spočívá v definování středů shluků, tzv. centroidů. Centroid je definovaný pro každý z K shluků, na které dělíme data. Metoda vždy rozdělí do K shluků všechny definované body, tedy žádný z bodů nezůstane ležet mimo klastr.

K-means rozděluje prostor podle Voroného diagramu. To znamená, že každému bodu b_m z množiny M (obsahující m prvků) je přidělena taková oblast V_m, že pro všechny body ležící ve V_m platí, že vzdálenost těchto bodů k b_m je menší než vzdálenost b_n, kde n se nerovná m [9]. Čili:

Mějme množinu bodů $M = \{b_1, b_2, ..., b_m\}$ ležících v prostoru R^N:

$\forall b_m \exists V_m \in R^N; x \in V_m \wedge k \neq m : |x - b_m| < |x - b_k|$

[1] Termín „K-means“ neboli K-průměr byl poprvé použit v roce 1967 Jamesem MacQueenem. Myšlenka však sahá až do roku 1957 a práce Hugo Steinhause jako technika pro pulse-code modulaci. Tato práce byla ovšem publikována Bellovými laboratořemi až roku 1982. Účinnější verze byla navržena a implementována do Fortranu Hartiganem a Wongem v roce 1975.



Obrázek 3.1: Voroného diagram pro 15 náhodných bodů v $x \in \{-4; 4\}$ a $y \in \{-4; 4\}$

Před startem algoritmu je nutné určit počet shluků K. Ten určuje i počet středů shluků, které jsou nutné k inicializaci algoritmu. Počáteční středy shluků můžeme zvolit buď zcela náhodně, nebo vybrat K bodů, nebo je můžeme zadat ručně (například pokud máme předpokládanou polohu středu shluků). V každém iteračním kroku algoritmus vezme všechny body, které se nacházejí v dané oblasti Voroného diagramu (podle středů), tzn. všechny body, pro které je daný střed zároveň nejbližším středem, a určí jejich průměr (například aritmetický průměr souřadnic). Tímto způsobem vznikne K nových středů a pak lze přejít stejným způsobem k další iteraci. V každém iteračním kroku je vypočteno K nových středů, dokud není průběh algoritmu přerušen zastavovací podmínkou, nebo se středy v dalším kroku už nijak nepohnou [2]. Zastavovací podmínkoui může být velikost posunu středů mezi iteračními kroky nebo určitý počet iterací.

[2]Je dokázáno, že algoritmus skončí, protože existuje jen konečný počet přiřazení bodů ke středům. Protože každá iterace vylepšuje výsledek, žádná z konfigurací se nebude opakovat. Proto se algoritmus vždy ustálí na nějaké konfiguraci.

Přestože algoritmus vždy skončí po určitém počtu kroků v závislosti na zvolených počátečních středech, nedosáhne vždy globálního optima. Proto naším dalším požadavkem je zpřesnění výsledků algoritmu. Nechceme-li kvůli tomu nadále zpřísňovat zastavovací podmínku (například proto, že při výpočtech s malým čísly se dopouštíme chyby při operacích s čísly definovanými pomocí plovoucí desetinné čárky), můžeme například spustit algoritmus vícekrát z různých počátečních středů a výsledky shlukování potom průměrovat. Dalším zlepšením, které však bylo zatím ověřeno jen experimentálně, je převedení dat v n-rozměrném prostoru do euklidovské koule [3] [4].

3.1.2 PCA

Principal component analysis [4], či Analýza hlavních komponent je matematická transformace sloužící k dekorelaci dat. Obvykle se využívá k snížení dimenze vektorů za předpokladu, že ztráta informací bude co nejmenší. Tato metoda se používá v rozpoznávání (feature extraction), ale také třeba ke kompresi dat. Pokud ji používáme ke shlukové analýze, je nutné po transformaci využít ještě nějaký shlukovací algoritmus (v případě naší aplikace K-means, který je popsán výše). Jelikož se jedná o velmi obecný algoritmus, jeho modifikace v závislosti na oblasti jejich aplikace jsou nazývany různě, například diskrétní Karhunen–Loèveho transformací (KLT), Schmidt–Mirskyův teórém, empirical orthogonal functions (EOF), eigenvalue decomposition (EVD), nebo faktorová analýza.

Před zahájením algoritmu je třeba si připravit data. V našem případě to znamená sadu příznakových vektorů (například systém bag of words). V každém rozměru se pak od každého prvku odečte průměr prvků rozměru. Po úpravě dat vypočteme kovarianční matici a její vlastní čísla a vlastní vektory. Takto spočítané vlastní vektory nejenom že jsou jednotkové (mají normu = 1), ale také jsou na sebe všechny kolmé. Ale důležitější je fakt, že vlastní vektory procházejí středy shluků a podávají nám tím důležité informace o vlastnostech shluků.

[3] Prostor euklidovské koule znamená, že vzdálenost všech shlukovaných bodů od počátku souřadného systému je menší než 1.

[4] PCA byla vynalezena v roce 1901 Karlem Pearsonem, jako obdoba věty hlavních os v mechanice. Nezávisle na Pearsonově výzkumu byla později vyvinuta a pojmenována Haroldem Hotellingem v 30. letech 20. století.

V tuto chvíli přichází na řadu snižování dimenze. Seřadíme-li vlastní vektory podle velikosti jejich vlastní čísel, získáme řadu seřazenou podle důležitosti vektorů. Obecně lze říci, že chceme-li počet dimenzí snížit o x, ignorujeme x vlastních vektorů náležících x nejmenším vlastním číslům. Navíc čím menší vlastní číslo je, tím méně informace jeho ignorováním ztrácíme.

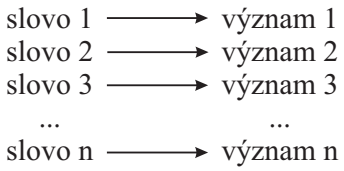
Poslední krok je opětovná transformace do m rozměrných příznakových vektorů. Původní dimenze n a nová dimenze m se mohou rovnat. Po výběru správných vlastních vektorů z nich vytvoříme matici tak, že všechny vektory transponujeme (tedy nejdůležitější vektor zůstává na prvním řádku). Čili:

$$[\text{ příznakový vektor }]_m = [\text{ matice vlastních vektorů }]^T \cdot [\text{ původní data }]_n^T$$

Vyřešením této banální rovnice získáme ideální transformaci z n do m-rozměrného prostoru [5].

3.1.3 LSA

Latentní sémantická analýza, nebo také Latentní sémantické indexování (LSI), je technika zpracování přirozeného jazyka. LSA doslovně znamená analýzu dokumentu za účelem nalezení jeho základního významu. Pokud by právě jedno slovo znamenalo právě jeden pojem, jednalo by se o triviální problém, stačilo by pouhé mapování relace mezi slovem a jeho významem.

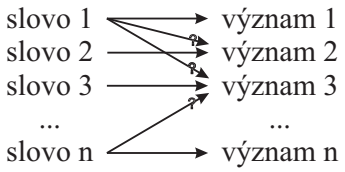


Obrázek 3.2: Jednoduchá významová mapa

V jazyce se však vyskytuje velké množství synonym (slova nebo slovní spojení se vzájemně stejným nebo podobným významem, která lze za určitých okolností zaměňovat),

homonym (slovo s totožnou grafickou podobou, ale rozdílným významem), frazémů (ustálené spojení slovních tvarů slov s vlastním významem, např. sousloví, přirovnání, atp.).

Kvůli těmto změnám ve významech slov je vytvoření relací mnohem obtížnější. Je k němu zapotřebí zohlednit vztah slova k ostatním výrazům ve větě, odstavci či článku. Vezměme si například slovo spojka. Vyskytuje-li se ve větě se slovy jako podstatné jméno, přídavné jméno, zájmeno, atd. jedná se pravděpodobně o spojku jako název slovního druhu. Pokud se však vyskytuje mezi výrazy jako rozptylka, čočka, paprsek, mluvíme pravděpodobně o spojce z optiky. Spojka stejně tak může být také označení pro mechanickou součástkou, styčnou osobou ve špionáži, nebo hráčskou funkcí v baseballu či softballu.



Obrázek 3.3: Složitá významová mapa

Základním přepokladem je, že slova podobného významu se objevují v podobných částech textu. LSA [5] se snaží přiblížit se od grafického zobrazení slova k jeho skutečnému významu tak, že zanese slovo do prostoru a dělá zde srovnání. Tento princip je reprezentován vytvořením matice, která je určena počtem výskytu slov v jednotlivých odstavcích popř. jiných úsecích textu. Řádky představují jednotlivá slova a sloupce jednotlivé odstavce, neboli reprezentace typu „bag of words“ (pytel slov). LSA odfiltruje některé šumy způsobené právě změnami významu. Zároveň se snaží nalézt minimální sadu pojmů, která se klene skrz všechny dokumenty. Problémem pak může být předpoklad, že každé slovo má právě jeden význam.

Ve většině složitějších systémů se používají ještě váhové funkce. Obvyklá úprava je zvednout váhu slov, která se vyskytují v malém počtu dokumentů, neboť je pravděpo-

[5] LSA byla patentována S. Deerwesterem, S. Dumaisovou, G. Furnasem, R. Harshmanem, T. Landauerem, K. Lochbaumovou a L. Streeterovou 15. září 1988. To znamená, že patentová ochrana vypršela teprve před necelými pěti lety.

dobnější, že slova, která se napříč články vyskytují málo mají pro ně větší význam. Jeden z algoritmů, který se pro tuto úpravu využívá (a je použit i v experimentu) se nazývá TF-IDF. Název TF-IDF je zkratka odvozená od dvou termínů. Term Frequency - četnost slova v dokumentu a Inverse document frequency - převrácená četnost slova ve všech dokumentech. TF složka vyjadřuje, kolikrát se výraz vyskytuje v dokumentu. Obvykle se normalizuje vydělením délkou (počtem slov) dokumentu, aby se předešlo nadhodnocování dlouhých dokumentů, ve kterých se hledaný výraz může vyskytovat častěji než v kratších. Tím získáváme následující definici TF:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3.1}$$

kde $n_{i,j}$ je počet výskytů slova t_i v dokumentu d_j. Jmenovatel reprezentuje součet počtu výskytů všech slov v dokumentu d_j , tj. jeho délku.

IDF složka reprezentuje „důležitost“ slova. Čím častěji se slovo vyskytuje v dokumentech, tím méně je důležité (slovo, které se vyskytuje ve všech dokumentech, jako například v angličtině člen „the“ nebo česká spojka „a“, je většinou pro vyhledávání nepoužitelné). IDF pro slovo spočítáme podle vzorce:

$$\mathrm{IDF_i} = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \tag{3.2}$$

kde $|D|$ je velikost databáze dokumentů, tedy počet dokumentů, ve kterých hledáme a $|\{j : t_i \in d_j\}|$ je počet dokumentů, které obsahují slovo t_i.

Příklad: Mějme sadu 10.000 dokumentů. Ve zkoumaném dokumentu, který má sto slov se termín „vzdělání“ vyskytuje třikrát. TF termínu vzdělání je tedy 3/100 = 0,03. Zkoumané slovo lze nalézt však pouze v deseti dokumentech, IDF je tedy log10(10000/10) = 4. Výsledné TF-IDF termínu „vzdělání“ je tedy součinem TF*IDF = 0,03*4 = 0,12. Dejme tomu, že slovo „a“ se v našem dokumentu objevilo například šestkrát, ale nalezneme jej v 9.900 ze všech dokumentů. TF-IDF pro slovo „a“ je tedy TF je 0,06 a IDF přibližně 0,0436. TF-IDF(a) = 0,06*log(...) = 0,00026. Využitím této váhové funkce je tedy jasné,

že termín „a“ je pro význam dokumentu nedůležitý (blízký nule). Díky velmi malé složce IDF lze předpokládat, že je nedůležitý i pro zbylý soubor dokumentů. Oproti tomu termín „vzdělání“ je pro dokument klíčový [8].

K sestrojení LSA se používá matematický postup zvaný singulární rozklad (singular value decomposition, popř. SVD). Díky využití SVD je snížený počet sloupců i při zachování informace o podobnosti mezi řádky. Podobnost slov je dána kosinem úhlu, popř. normovaným skalárním součinem mezi libovolnými dvěma vektory (řádky). Údaj blížící se k 1 znamená velkou podobnost, je-li údaj blízký 0 pak se jedná o velmi malou podobnost.

Důvodem, proč je SVD tolik užitečné, je to, že nám může pomoci najít sníženou dimenzionální reprezentaci naší matice, která potom klade důraz na nejsilnější vztahy a odstraní šum. Trik při použití SVD je přijít na to, kolik rozměrů použít při aproximaci. Málo důležité rozměry vnášejí šum, a proto je vhodné je vynechat [6, 11].

3.2 Metody zhodnocení kvality shlukování

3.2.1 Homogeneity (Stejnorodost)

Homogeneity nabývá hodnoty od 0 do 1. Výsledek shlukování je stejnorodý, tedy roven 1, pokud všechny prvky jedné třídy leží v jednom shluku. Nezáleží na pořadí prvků vektorů výsledků shlukování a správného řešení (tzn. všechny permutace by měly mít stejný výsledek). Jak moc se od sebe liší, popisuje podmíněná entropie, z anglického conditional entropy. Je-li homogenita 1, je conditional entropy 0. Homogeneity h je dána jako:

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{3.3}$$

kde $H(C|K)$ je podmíněná entropie tříd přiřazených shluku a je dána vztahem:

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n} \cdot log(\frac{n_{c,k}}{n}) \tag{3.4}$$

a H(C) je entropie tříd a je dána vztahem:

$$H(C|K) = \sum_{c=1}^{|C|} \frac{n_c}{n} \cdot log(\frac{n_c}{n}) \tag{3.5}$$

s celkovým počtem vzorků n, n_c a n_k je počet vzorků tj. počet vzorků patřící do třídy C a přiřazených do k-tého clusteru, a konečně $n_{c,k}$ je počet vzorků třídy C, ale přiřazených shluku k.

Podmíněná entropie klastrů dané třídy $H(K|C)$ a entropie klastrů $H(K)$ jsou definovány symetrickým způsobem[12].

3.2.2 Completeness (Kompletnost)

Completeness nabývá hodnoty od 0 do 1 od 0 do 1. Výsledek je kompletní, tedy roven 1, pokud jsou shluknuté všechny prvky a zároveň se shluky dokonale „pasují“ do tříd. Podobně jako u homogenity se jejich rozdíl počítá z podmíněné entropie. Completeness c je dána jako[12]:

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{3.6}$$

3.2.3 V-measure

V-measure nabývá také hodnot od nuly do jedné, neboť se jedná o harminický průměr Homogeneity a Completeness. Lze jej tedy určit následovně[12]:

$$v = 2 \cdot \frac{h \cdot c}{h + c} \tag{3.7}$$

3.2.4 Adjusted Rand Index

Aby bylo možné porovnávat výsledky shlukování vůči vnějším kritériím, je nutná míra shody. Jelikož se předpokládá, že každý prvek je přiřazen pouze jedné třídě v oblasti vnějšího kritéria pouze v jednom klastru, lze použít míru shody mezi těmito dvěma oddíly.

Mějme sadu n prvků $S = O_1, O_2, ..., O_n$ a předpokládejme, že $U = u_1, u_2, ..., u_n$ a $V = v_1, v_2, ..., v_n$ představují dva různé oddíly objektů S tak, že $\cup_{i=1}^{R} u_i = S = \cup_{j=1}^{C} v_j$ a $u_i \cap u_{i'} = \emptyset = v_j \cap v_{j'}$ pro $1 \leq i \neq i' \leq R$ a $1 \leq j \neq j' \leq C$. U je tedy naše vnější kritérium a V jsou výsledky shlukovací analýzy. Považujme a za počet všech dvojic, které jsou ve stejném shluku v U i V.$_b$ je počet dvojic prvků, které jsou ve stejné třídě v U, ale v různých shlucích v V.$_c$ je počet dvojic, které jsou v různých třídách v U, ale rozdílných klastrech v V.$_d$ je počet dvojic, které jsou v rozdílných shlucích v U i ve V. Čísla a a b se pak nazývají shody, zatímco c a d se nazývají neshody. Rand index [6] lze pak spočítat jako:

$$RI = \frac{a + d}{a + b + c + d} \tag{3.8}$$

Problémem Rand indexu však je, že hodnota RI u dvou náhodných oddílů není konstatní hodnota (řekněme nula). Adjusted Rand index [7] předpokládá generalizované hypergeometrické rozdělení jako model náhody. Oddíly U a V jsou vybrány tak, že počet objektů ve třídách a shlucích je pevný. Nechť n_{ij} je počet objektů, které jsou zároveň ve třídě n_i a ve shluku n_j. $n_{i.}$ a $n_{j.}$ je počet objektů v třídě u_i a shluku v_j.

Obecná forma indexu s očekávanou konstatní hodnotou je:

$$\frac{index - očekávanýindex}{maximálníindex - očekávanýindex} \tag{3.9}$$

která je omezená shora 1 a má hodnotu 0, pokud se index rovná jeho očekávané hodnotě. V rámci obecného hypergeometrického modelu lze dokázat, že:

$$E\left[\sum_{i,j} \binom{n_{ij}}{2}\right] = \frac{\left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right]}{\binom{n}{2}} \tag{3.10}$$

[6] Rand index zveřejnil W. M. Rand v publikaci „Objective criteria for the evaluation of clustering methods“ v roce 1971.

[7] Adjusted Rand index byl vynalezen až roku 1985 Lawrencem Hubertem and Phippsem Arabiem

Výraz $a + b$ může být zjednodušen na lineární transformaci $\sum_{ij} \binom{n_{ij}}{2}$. Jednoduchou algebrou lze vzorec pro výpočet Adjusted Rand indexu zjednodušit [3, 7]:

$$ARI = \frac{\left[\sum_{i,j} \binom{n_{ij}}{2}\right] - \frac{\left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right]}{\binom{n}{2}}}{\frac{\left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right]}{2} - \frac{\left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right]}{\binom{n}{2}}} \tag{3.11}$$

Kapitola 4

Experimentální část

Tato část práce se věnuje testování jednotlivých algoritmů na vzorových vstupech. Každá z metod byla testována na každém ze tří vstupních souborů. Proto je experimentální část práce rozdělena do tří oddílů, který každý popisuje chování algoritmů na různém typu vstupních dat. Na dalších stránkách jsou podrobně popsány tyto tři experimenty:

1. 162 novinových článků v českém jazyce. Články jsou rozděleny do devíti nerovnoměrných tříd. Cílem experimentu je ozkoušet algoritmy na reálném problému.

2. 35 recenzí na různé výrobky v anglickém jazyce. Články jsou rozděleny do sedmi rovnoměrných tříd po pěti článcích. Cílem experimentu je ozkoušet správnost algoritmů.

3. 200 novinových článků v českém jazyce v lemmatizované a nelemmatizované verzi. Články jsou rozděleny do deseti rovnoměrných tříd po dvaceti článcích. Cílem experimetu je porovnat výsledky algoritmů při lemmatizovaném [1] / nelematizovaném vstupu.

[1] To znamená, že slova jsou převedena do základního tvaru, který se nazývá „lemma“. Více informací v kapitole 4.3

Struktura každého experimentu je stejná. Algoritmy byly testovány v tomto pořadí:

1. Hashing Vectorizer + **K-means**
2. TF-IDF + **K-means**
3. TF-IDF + **PCA** + K-means
4. **LSA** + K-means

Nejprve byl u každého z algoritmů proveden test na optimální nastavení, popřípadě test na ideální redukci dimenzí. Tento test byl vyhodnocen na základě grafu. Další test je zaměřený na nastavení správného počtu klastrů.

4.1 1. Experiment

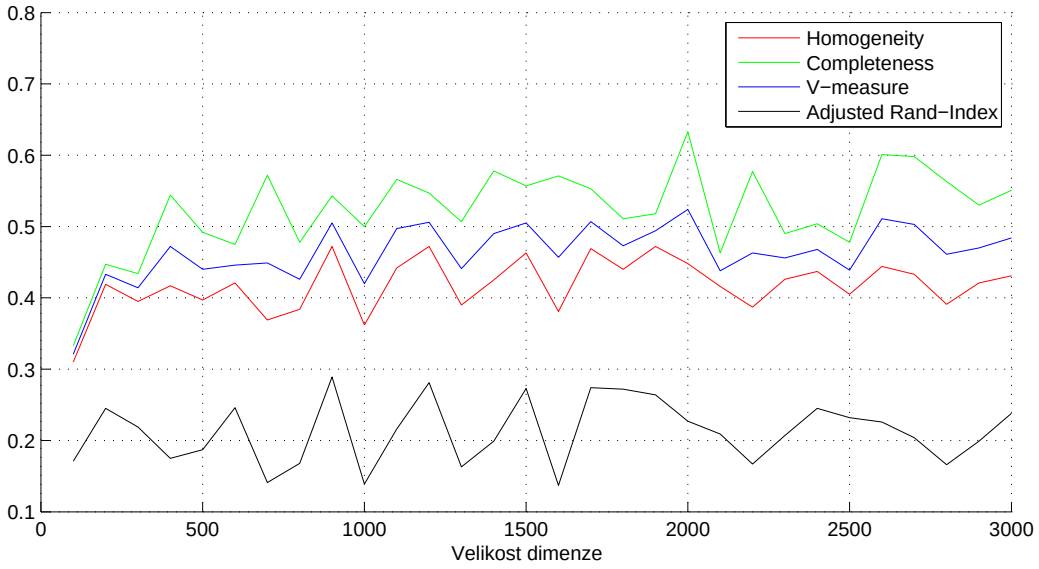
Jak již bylo zmíněno, první experiment proběhl se sadou 162 novinových článků v českém jazyce. Články jsou rozděleny do devíti nerovnoměrných tříd. Cílem experimentu je ozkoušet algoritmy na reálném problému.

4.1.1 Hashing Vectorizer + K-means

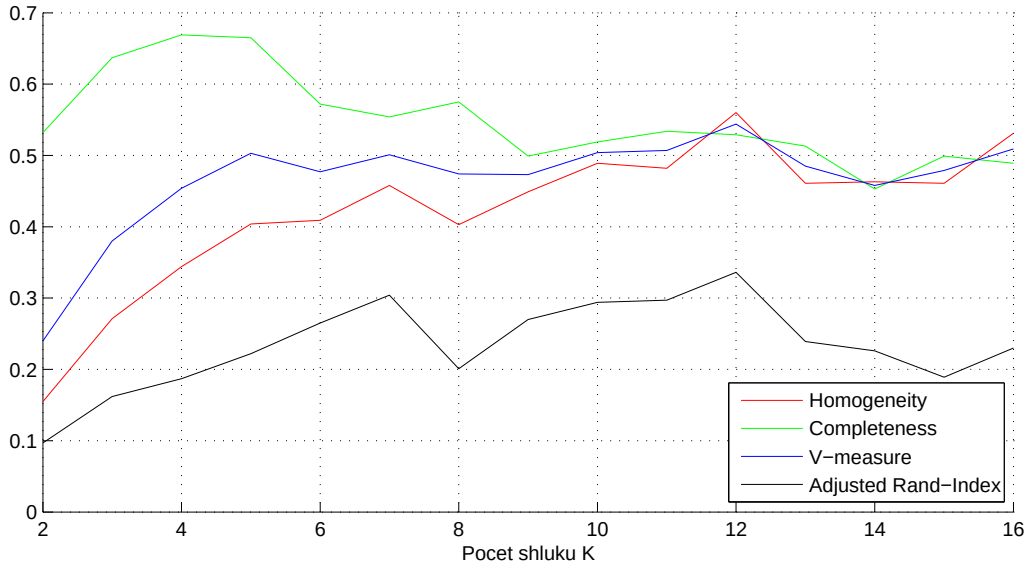
Pro dosažení lepších výsledků při používání K-means (a při jeho dalším využívání v kombinaci s ostatními metodami) jsem použil dvě úpravy. Za prvé velice malá zastavovací podmínka znamená, že iterování se zastaví až ve chvíli, kdy se středy shluků téměř nemění. Za druhé výpočet proběhne desetkrát a výsledek je průměrem všech vypočtených výsledků. S těmito úpravami trval výpočet přibližně 1,5 sekundy, přičemž časově nejnáročnější je jednoznačně vícenásobný běh algoritmu.

Prvním použitým algoritmem je Hashing Vectorizer, který transformuje vektory vzniklé bag of words systémem do dané dimenze a ty pak ještě vydělí tak, aby ležely v euklidovské kouli (euklidovská vzdálenost od středu <= 1), což zlepšuje výsledky K-means, jak je zmíněno výše.

První test má určit ideální nastavení redukce dimenzí. Protože výpočty nebyly časově náročné, byl test proveden podle poměrně jemného dělení a to pro násobky sta dimenzí od nastavení $D = 100$ po $D = 3000$. Cílem je najít oblast s výrazně lepšími výsledky v hodnocení kvality. Na základě tohoto testu je v dalším pokračování experimentu použita transformace do dimenze $D = 1200$, která prokazovala nejlepší výsledky z testovaných nastavení. Dimenze je téměř čtyřikrát menší oproti TF-IDF, kde je D=4719.



Obrázek 4.1: 1. Experiment - Test nastavení Hashing Vectorizeru

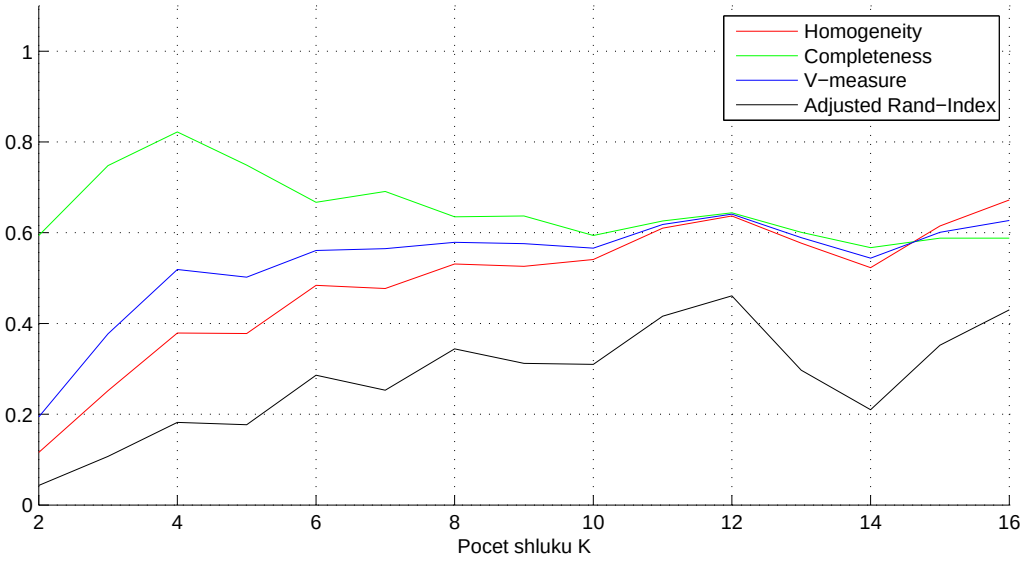


Obrázek 4.2: 1. Experiment - HV + K-means - test nastavení K

Jak je vidět z grafu, nejlepší výsledky vykazuje shluknutí do sedmi, popřípadě dvanácti kategorií. Chyb může být několik. Problémem může být špatně zvolená váhová funkce, potíže s příbuznými slovy v češtině (tenista - tenistka, atp.), ale také nepřesné roztřídění článků, podle kterých analýzu výsledků provádíme.

4.1.2 TF-IDF + K-means

TF-IDF je jedna z obvyklých úprav dat, tzv. preprocesingu. Při ní je zvednuta váha slov, která se vyskytují v malém počtu dokumentů. Je totiž pravděpodobnější, že slova, která se v článcích vyskytují málo mají větší význam. Jak již bylo zmíněno výše, po TF-IDF transformaci zůstává 4719 znaků pro každý dokument.



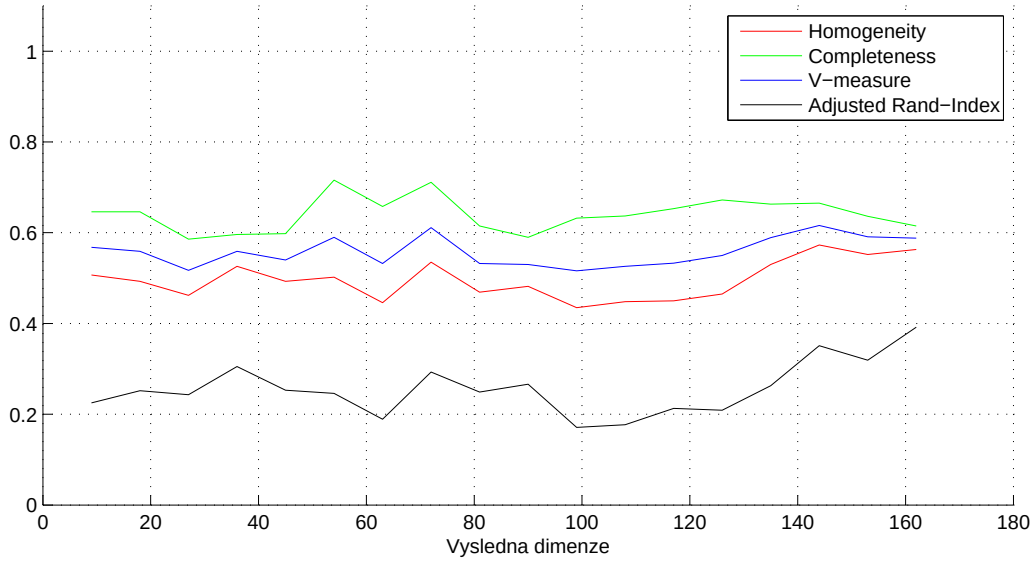
Obrázek 4.3: 1. Experiment - TF-IDF + K-means - test nastavení K

Také druhý pokus s algoritmem K-means ukazuje, že nejlépe vypadá způsob rozdělení na dvanáct různých clusterů. To značně zmenšuje šanci, že chyba je zavedena špatně zvolenou váhovou funkcí.

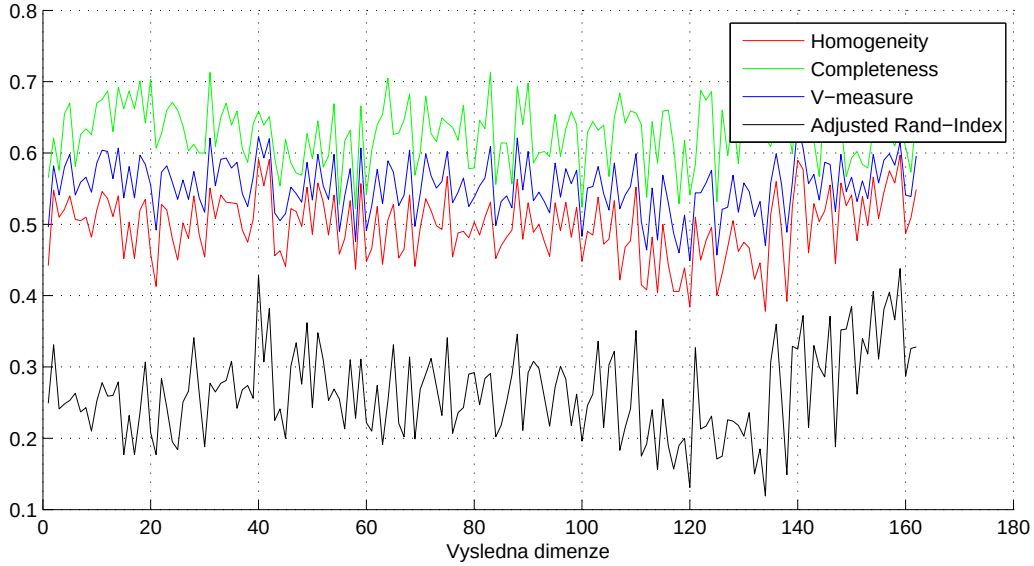
4.1.3 TF-IDF + PCA + K-means

Při nastavení PCA je důležitým krokem vybrání výsledné dimenze. Tu není možné nastavit vyšší, než je počet obrazů, čili 162. Díky dostatku výpočetního času (1 výpočet trval přibližně 0,5 s), bylo možno otestovat všechna nastavení od 1 do 162 dimenzí. V tabulce je druhý test z hrubším nastavením. (Test proběhl pro $K = 9$.)

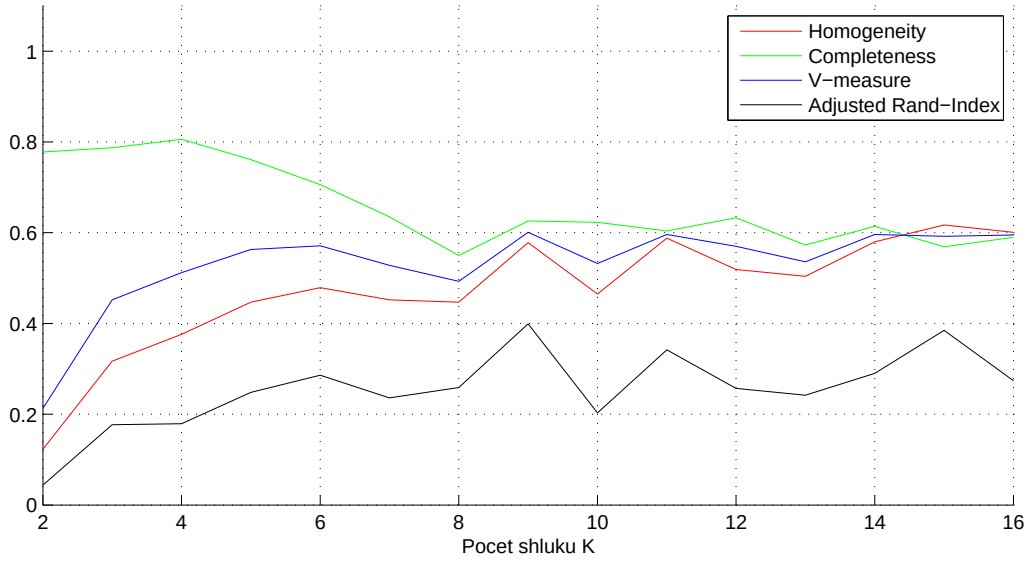
Protože výsledky z testu nastavení dimenze nejsou příliš jednoznačné, pro další testování bude PCA snižovat dimenzi na 162.



Obrázek 4.4: 1. Experiment - TF-IDF + PCA + K-means - test hrubé nastavení dimenze



Obrázek 4.5: 1. Experiment - TF-IDF + PCA + K-means - test jemné nastavení dimenze

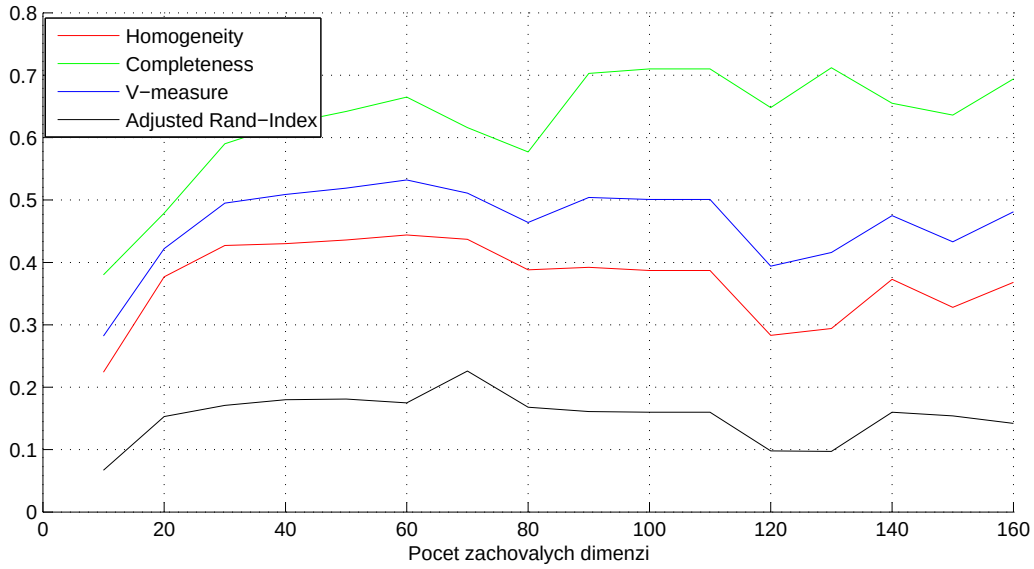


Obrázek 4.6: 1. Experiment - TF-IDF + PCA + K-means - test nastavení K

Test objevil dvě nápadně lepší shluknutí. První je pro $K = 9$, které jsme předpokládali a které také dosahuje nejlepšího výsledku. Druhým shluknutím s výrazně lepším ohodnocením je K=15. Zvláštním úkazem oproti ostatním testům je ale nižší index Completeness než Homogeneity.

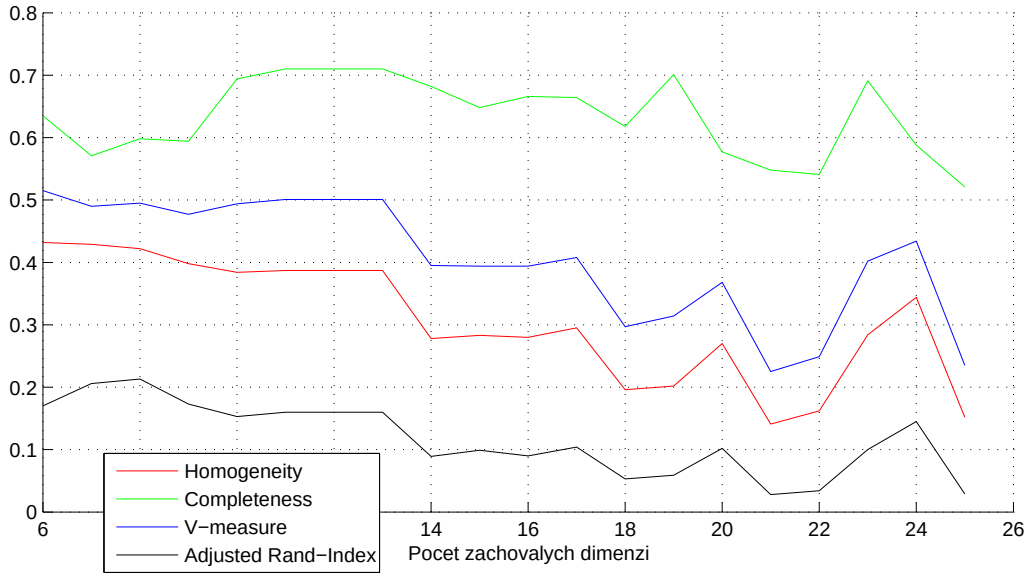
4.1.4 LSA + K-means

LSA je sice výpočtově daleko náročnější (1 výpočet trval téměř 22 s), ale zato nám poskytuje kromě informace o tom, kde se v prostoru nachází celý článek (obraz), obsahuje výsledek i informaci, kde se v prostoru nacházejí jednotlivá slova. To znamená, že můžeme sledovat relaci mezi slovem a článkem. Bohužel tato informace je pro účel rozdělení článků do kategorií na základě významů neužitečná, pokud bychom nechtěli kategorie „pojmenovávat“ automaticky pomocí slov blízkých článku. Protože trik u LSA je snížení dimenze prostoru dokumentů pomocí výběru největších singulárních čísel a výpočet je poměrně časově náročný, první test poslouží k hrubému odhadu, nakolik bude nejlepší dimenzi snížit. (Test proběhl pro $K = 9$.)

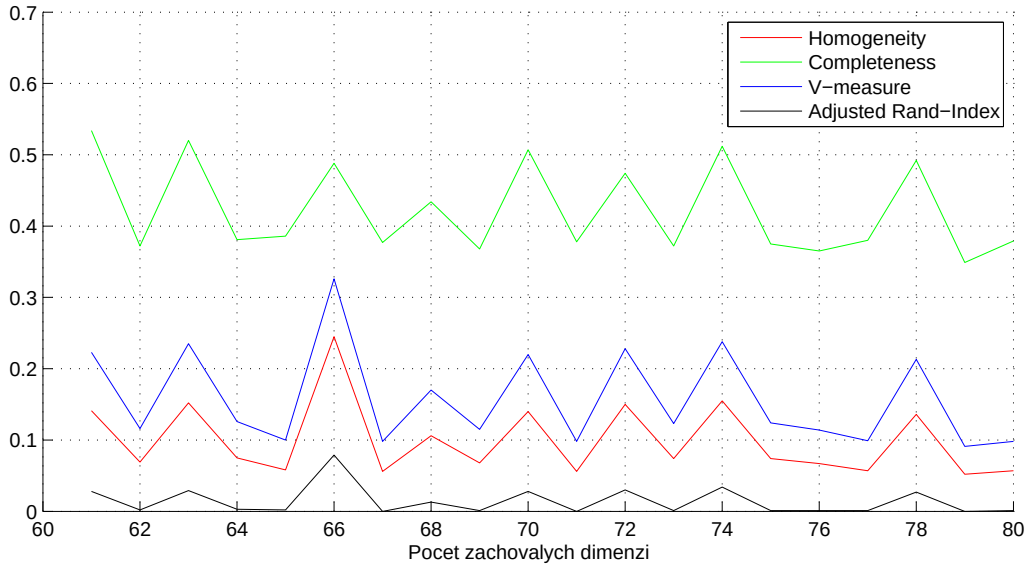


Obrázek 4.7: 1. Experiment - LSA + K-means - test hrubého nastavení D

Na základě hrubého vyšetření jsem se rozhodl zaměřit se přesněji na oblasti ponechaných dimenzí $6 - 25$ a $61 - 80$.

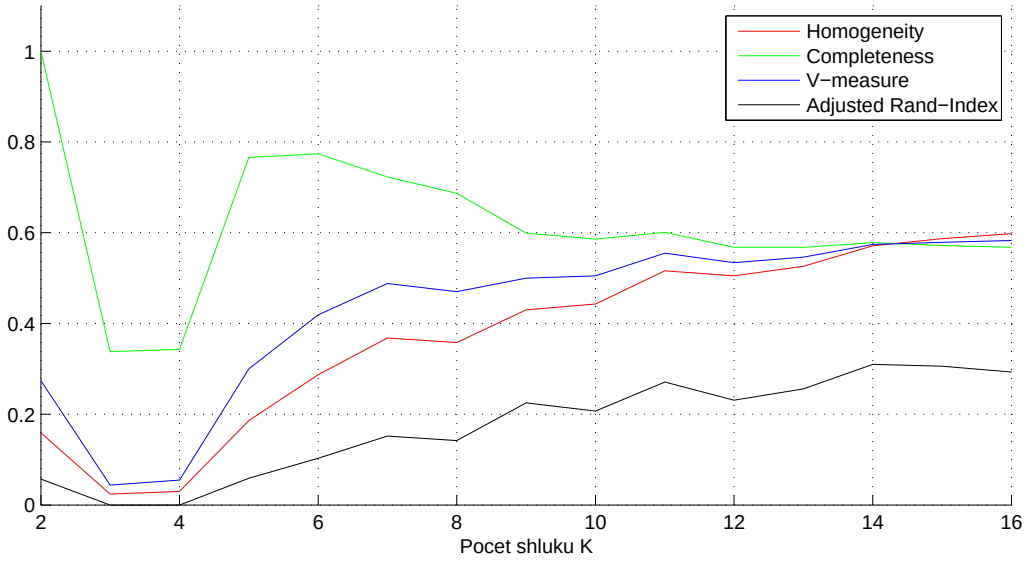


Obrázek 4.8: 1. Experiment - LSA + K-means - test nastavení D (6-25)



Obrázek 4.9: 1. Experiment - LSA + K-means - test hrubého nastavení D (61-80)

Přesnější pokus ukázal, že na ponechání více dimenzí nemá význam se soustředit a proto pokračování provedeme na prostoru dimenze 8. Na výsledek LSA jsem aplikoval stejný test rozdělení pomocí K-means pro $K = 2, 3, ..., 16$.



Obrázek 4.10: 1. Experiment - LSA + K-means - test nastavení K

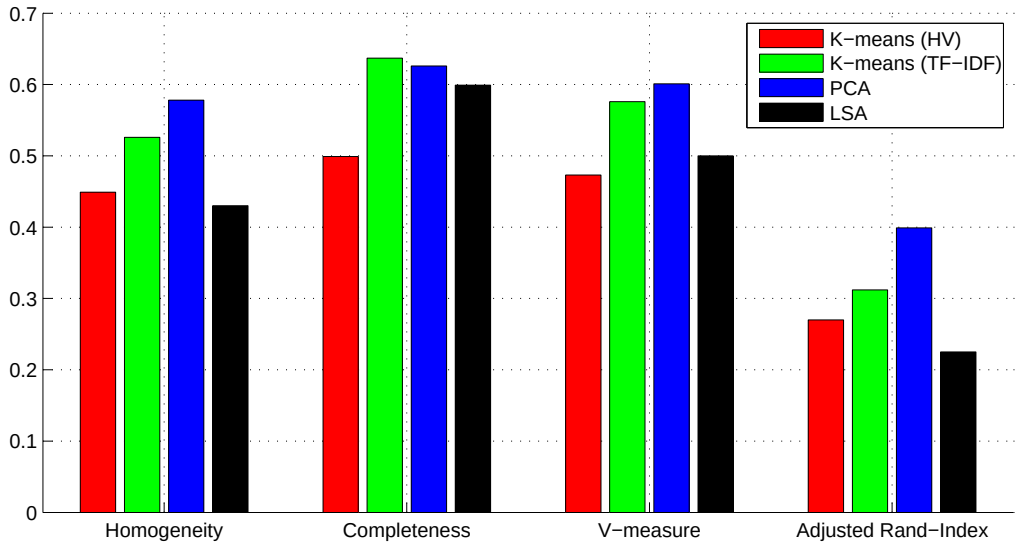
4.1.5 Shrnutí

Porovnání kvality algoritmů u prvního pokusu jsem provedl na základě dvou hlavních kritérií. První kritérium vychází z předpokladu, že správným výsledkem je 9 shluků. Druhým kritériem je kvalita samotná, ze všech pokusů je tedy vybrán ten s nejlepším výsledkem zcela nezávisle na tom, do kolika shluků obrazy rozdělil.

Hodnotíme-li výsledky na základě prvního kritéria, nejlepších dosáhla metoda PCA. To se však dalo předpokládat, neboť jako jediná z testovaných metod měla nejlepší výsledek právě pro nastavení $K = 9$.

Tabulka 4.1: 1. Experiment - Srovnání výsledků algoritmů pro K=9

typ algoritmu	Homogeneity	Completeness	V-measure	Adj. Rand-Index
K-means HV	0,449	0,499	0,473	0,270
K-means TF-IDF	0,526	0,637	0,576	0,312
PCA	0,578	0,626	0,601	0,399
LSA	0,430	0,599	0,500	0,225

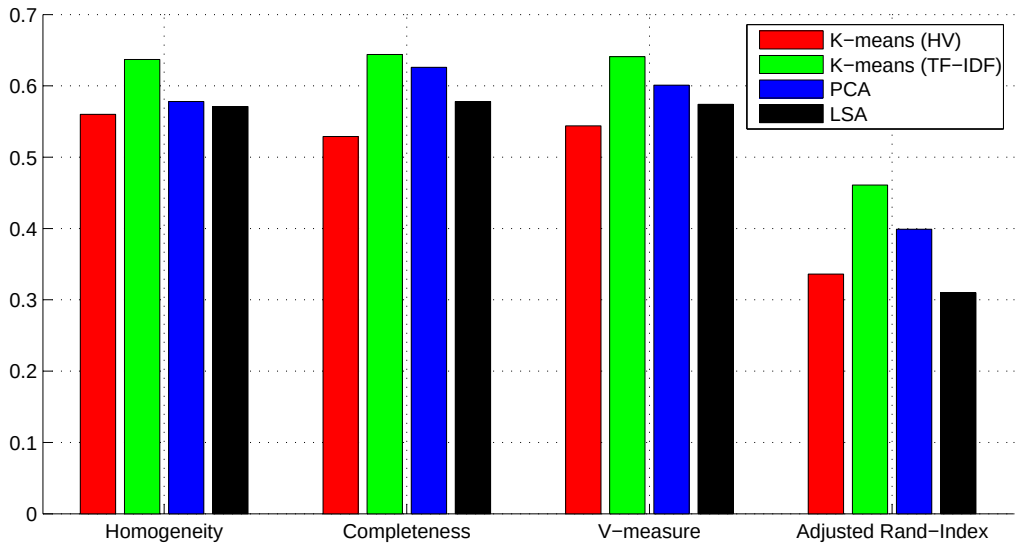


Obrázek 4.11: 1. Experiment - Srovnání výsledků algoritmů pro K=9

Pokud metody hodnotíme nezávisle na tom, do kolika clusterů prostor rozdělily, dosáhl nejlepších výsledků algoritmus K-means s TF-IDF váhovou funkci. Ten však dělil prostor do 12 shluků, stejně tak jako K-means s Hashing Vectorizerem. Nejhorších výsledků dosáhla LSA, která na druhou stranu je velmi citlivá na příbuzná slova, což by mohly prokázat či vyvrátit následující dva experimenty.

Tabulka 4.2: 1. Experiment - Srovnání výsledků algoritmů pro nejlepší K

typ algoritmu	Homogeneity	Completeness	V-measure	Adj. Rand-Index
K-means HV	0,560	0,529	0,544	0,336
K-means TF-IDF	0,637	0,644	0,641	0,461
PCA	0,578	0,626	0,601	0,399
LSA	0,571	0,578	0,574	0,310



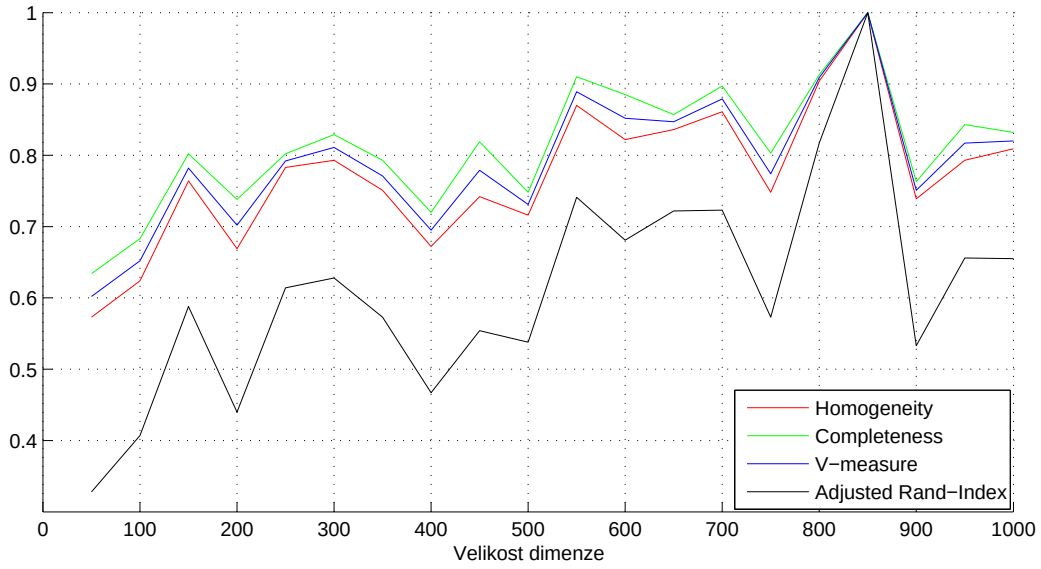
Obrázek 4.12: 1. Experiment - Srovnání výsledků algoritmů pro nejlepší K

4.2 2. Experiment

Druhý experiment byl více zaměřený na otestování funkčnosti jednotlivých algoritmů. Rozsah dokumentů je zde menší, jedná se o 35 článků v anglickém jazyce. Soubor tedy obsahuje celkem 7 kategorií po 5 článcích, z toho 30 recenzí na různé výrobky. Soustředil jsem se hlavně na poslední tři skupiny, které jsem schválně zvolil tak, aby používaly podobná slova. Články 21-25 jsou tedy recenze vysavačů, články 26-30 recenze robotických vysavačů a články 31-35 recenze čističů koberců (vysavačů určených pouze na koberce). Předpokladem tedy je, že pro $K = 7$ budou články rozděleny po pěti správně do svých náležitých kategorií. Pro $K = 6$ by měl jeden shluk vytvořit vysavače a robotické vysavače a konečně pro nastavení $K = 5$ by se v jednom shluku měly nacházet články 21-35, tedy posledních 15 článků.

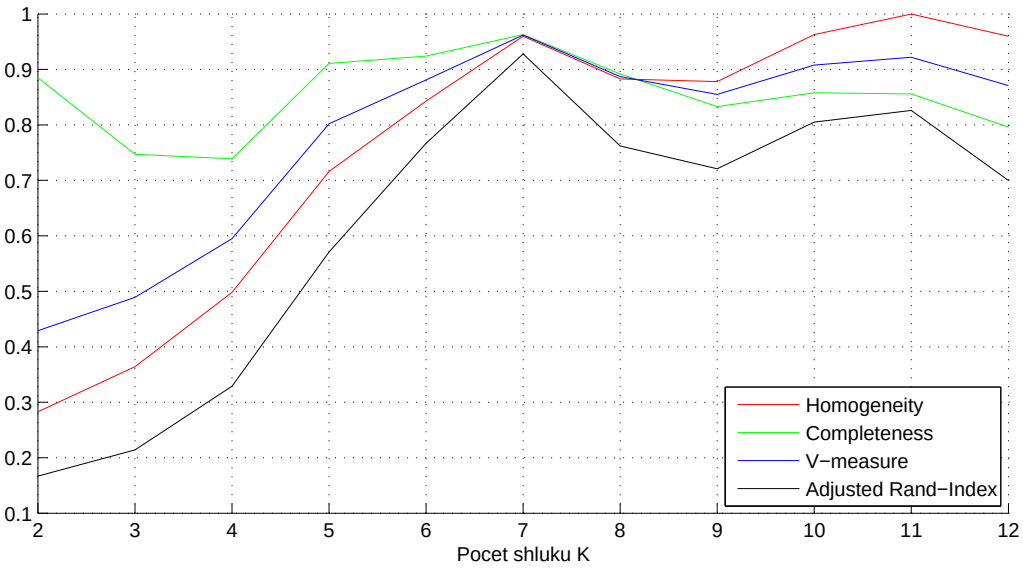
4.2.1 Hashing Vectorizer + K-means

Jednotlivé algoritmy jsem zkoušel ve stejném pořadí jako u předchozího experimentu. Protože slov je zde mnohem méně než v předchozím experimentu, volil jsem nastavení Hashing Vectorizeru daleko jemněji:



Obrázek 4.13: 2. Experiment - HV + K-means - Test nastavení snížení dimenze

Jelikož pro nastavení 850 dimenzí nastala situace, kdy shluková analýza dopadla přesně podle našeho odhadu (tj. všechna kritéria hodnocení kvality se rovnají jedné), bude pokus s nastavením K probíhat právě pro $D = 850$. Předpokladem však je, že stejně dobrého výsledku pro žádné jiné K dosáhnout nelze.

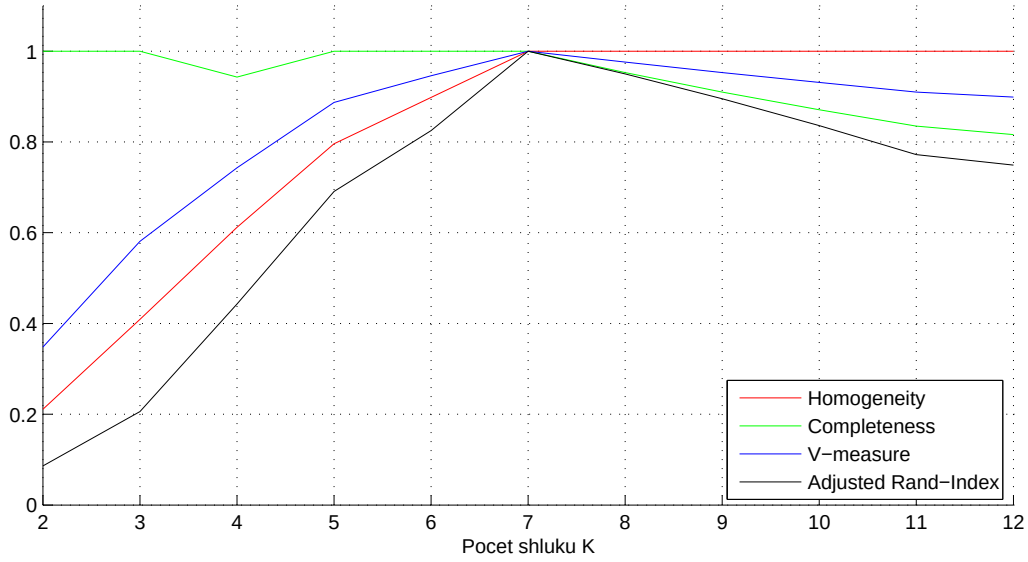


Obrázek 4.14: 2. Experiment - HV + K-means - Test nastavení K

Nejlepší výsledek byl nalezen pro nastavení $K = 7$, ale ukázala se jedna z vlastností algoritmu K-means. Ten totiž nevede vždy ke stejným výsledkům, proto v druhém pokusu neshlukl algoritmus dokumenty do sedmi stejně velkých kategorií, ale jeden z článků umístil jinam. Tím pádem je však vyvrácen náš předpoklad. Mohou nastat případy, kdy by jiná nastavení mohla dosáhnout lepších výsledků. Poměrně dobře tomu však můžeme předcházet průměrováním výsledků.

4.2.2 TF-IDF + K-means

Další pokus je TF-IDF transformace a K-means. Po aplikování zůstane celkem 519 parametrů. Následující graf ukazuje opět nastavení pro $K = 2, 3, ..., 12$.

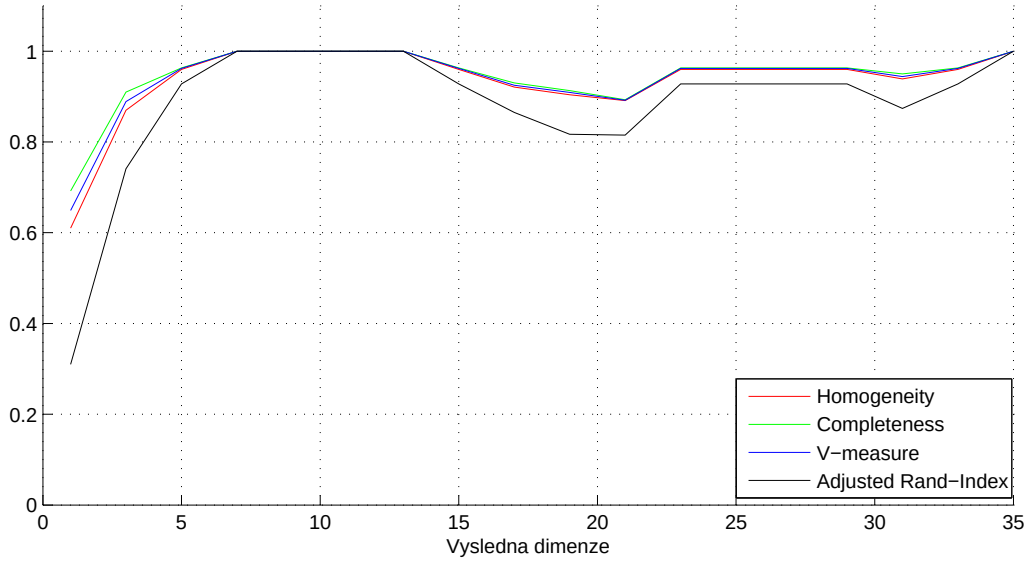


Obrázek 4.15: 2. Experiment - TF-IDF + K-means - Test nastavení K

Tento test jednoznačně ukázal, že nejpřesnější nastavení je $K = 7$. I při menším počtu výpočtů k průměrování nevykazoval nestability, což znamená značnou úsporu výpočetního času. Je tedy na první pohled vidět, že vhodně zvolená metoda předzpracování (preprocessingu) má veliký vliv na výsledek.

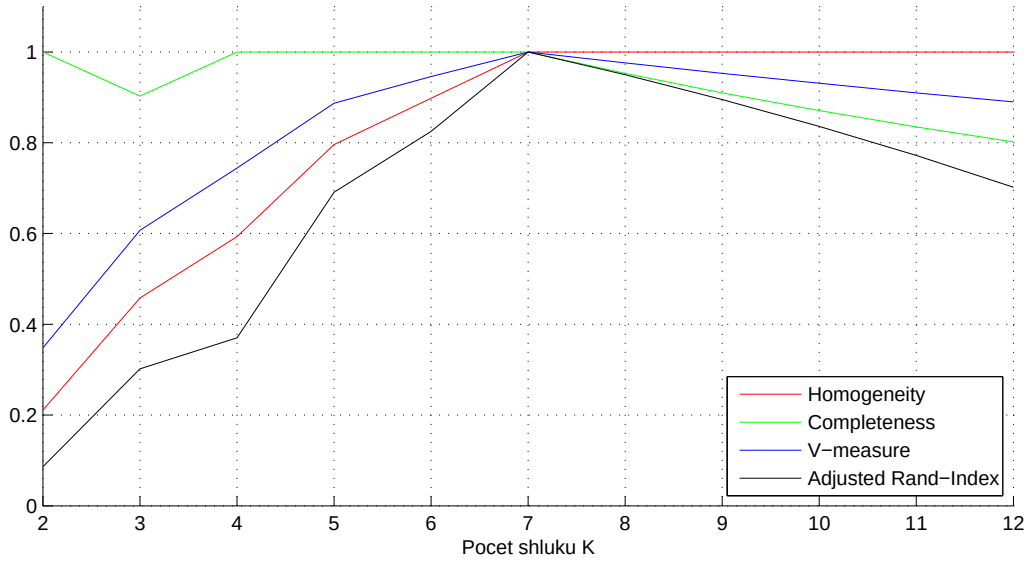
4.2.3 TF-IDF + PCA + K-means

Nastavení výsledné dimenze jsem si mohl dovolit testovat velice jemně, neboť rozsah dat je malý a maximální nastavení dimenze je 35.



Obrázek 4.16: 2. Experiment - TF-IDF + PCA + K-means - Test nastavení dimenze

Na základě testu se zdá, že oblast $D = 7, ..., 13$ by byla vhodná pro transformaci. Proto jsem pro další test použil střed tohoto prostoru, tedy $D = 10$.

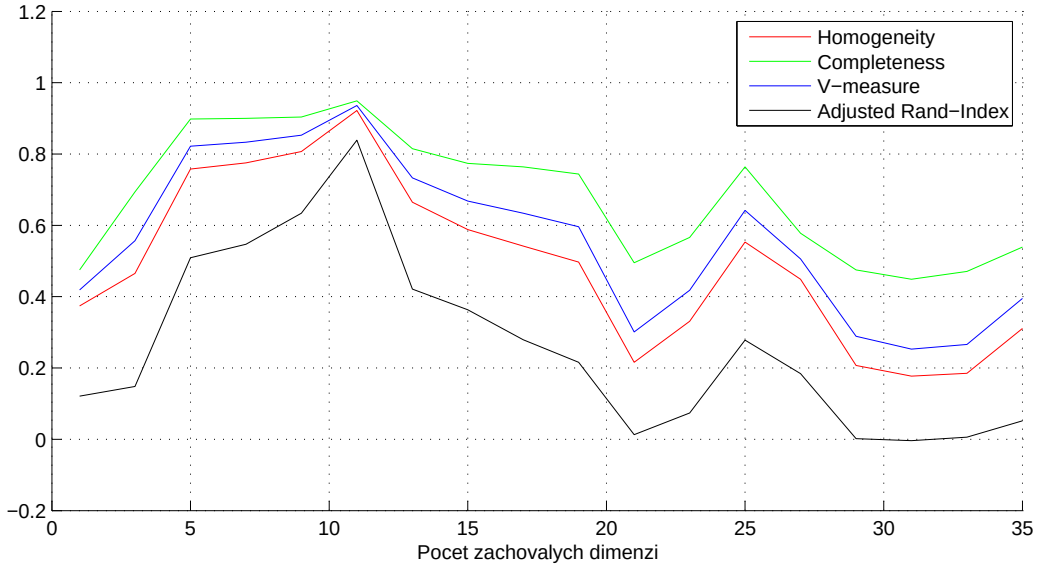


Obrázek 4.17: 2. Experiment - TF-IDF + PCA + K-means - Test nastavení K

Stejně jako předchozí test i kombinace metody transformace TF-IDF, PCA a K-means dosáhla předpokládaných výsledků, čili ideálnímu shluknutí při nastavení $K = 7$. Tato kombinace algoritmů byla dokonce ještě rychlejší než předchozí a navíc rozsah ideálního snížení dimenzí byl velký, tudíž méně náchylný ke špatné volbě.

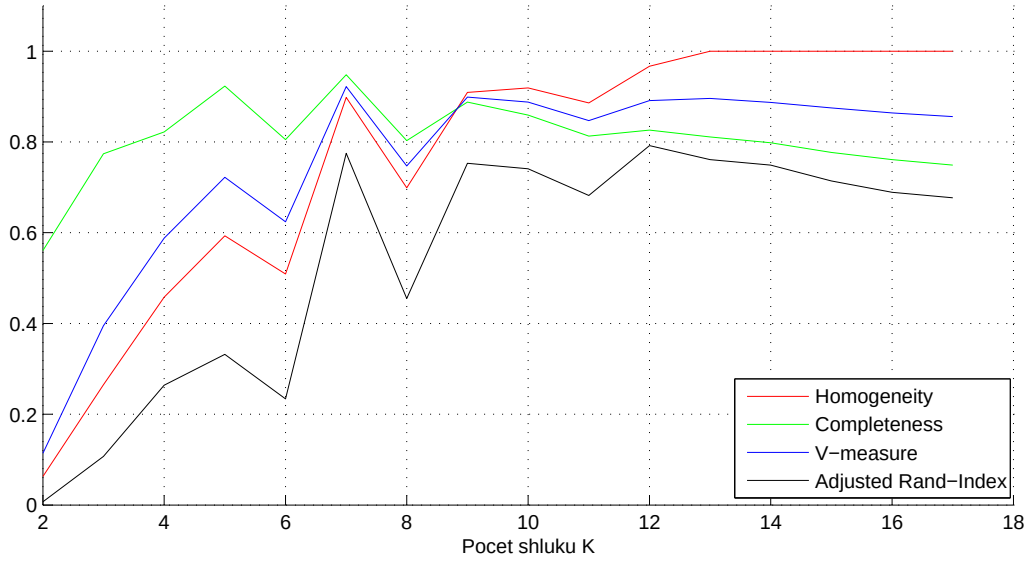
4.2.4 LSA + K-means

Prvním testem u LSA potřebujeme zjistit, nakolik je vhodné snížit dimenzi prostoru. Jelikož maximální nastavení je opět 35, prozkoumal jsem stejná nastavení jako u předchozí metody.



Obrázek 4.18: 2. Experiment - LSA + K-means - Test nastavení snížení dimenze

Protože výrazně nejlepších výsledků dosáhla redukce na 11 dimenzí, do dalšího testu jsem vybral pouze vektory odpovídající 11 největším singulárním číslům. Následující test ukázal nejlepší nastavení počtu clusterů. Kvůli dobrým výsledkům při větším množství shluků je testované nastavení pro $K = 2, 3, ..., 17$.



Obrázek 4.19: 2. Experiment - LSA + K-means - Test nastavení K

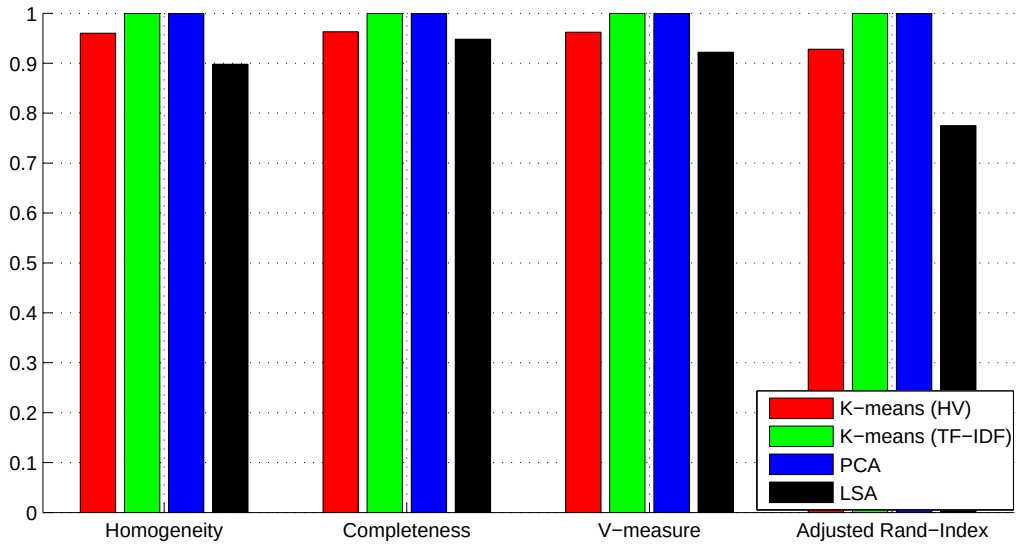
Výsledky pro nastavení $K = 7$ a $K = 12$ se jeví jako nejlepší. LSA však bohužel nedosáhla stejných výsledků jako předchozí algoritmy.

4.2.5 Shrnutí

Protože všechny algoritmy měly velice dobré, či dokonce nejlepší výsledky pro rozdělení do sedmi shluků, je srovnání provedeno právě pro $K = 7$. Nejrychlejší algoritmem byla kombinace TF-IDF transformace, PSA a K-means a zároveň měla skvělé výsledky. Druhý experiment byl však spíše „školním problémem“ jednak z hlediska rozsahu dat, jednak díky jednoznačnosti článků. Překvapivé bylo, že poměrně špatné výsledky vykazovala časově nejnáročnější Latentní sémantická analýza. To naznačuje, že jí nevyužíváme nejvhodnějším způsobem.

Tabulka 4.3: 2. Experiment - Srovnání výsledků algoritmů pro K=7

typ algoritmu	Homogeneity	Completeness	V-measure	Adj. Rand-Index
K-means HV	0,960	0,963	0,962	0,928
K-means TF-IDF	1,000	1,000	1,000	1,000
PCA	1,000	1,000	1,000	1,000
LSA	0,898	0,948	0,922	0,775



Obrázek 4.20: 2. Experiment - Srovnání výsledků algoritmů pro K=7

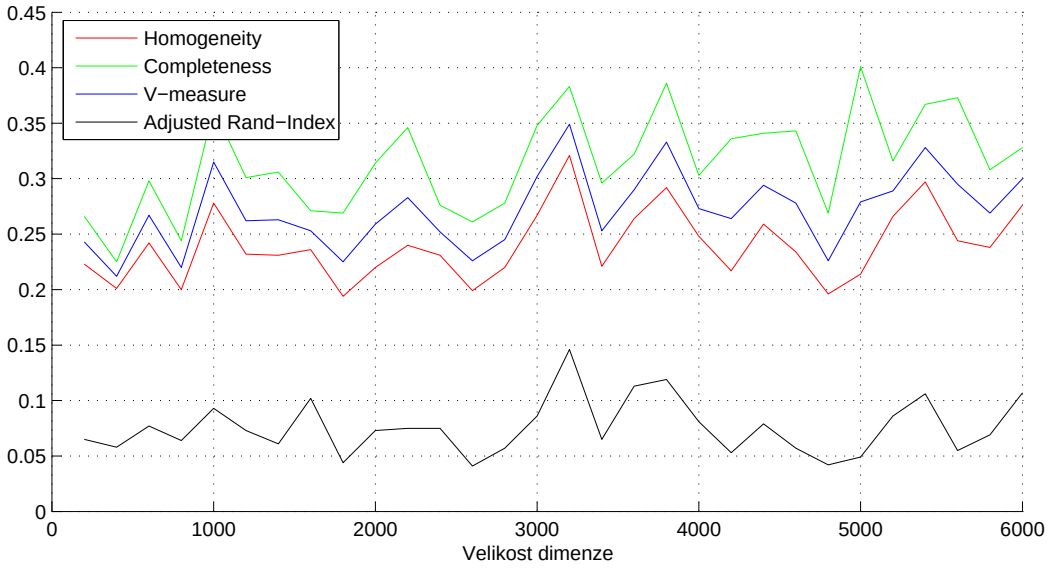
4.3 3. Experiment

Poslední experiment má za účel prověřit rozdíl mezi sadou česky psaných dokumentů jednak v neupravené verzi, jednak v lemmatizované verzi. To znamená, že slova jsou předělána na základní tvar, který se nazývá „lemma“. Čili například místo slova „prosince“ bude v dokumentu slovo „prosinec“. Soubor se skládá z 200 článků rozdělených pravidelně do deseti různých kategorií po dvaceti článcích. Porovnávaný soubor obsahuje stejné články ve stejném pořadí v lemmatizované verzi.

Protože pro srovnání potřebujeme opravdu stejné podmínky, budu správnost nastavení vždy testovat na původních článcích a stejné nastavení pak bude aplikováno i na články lemmatizované.

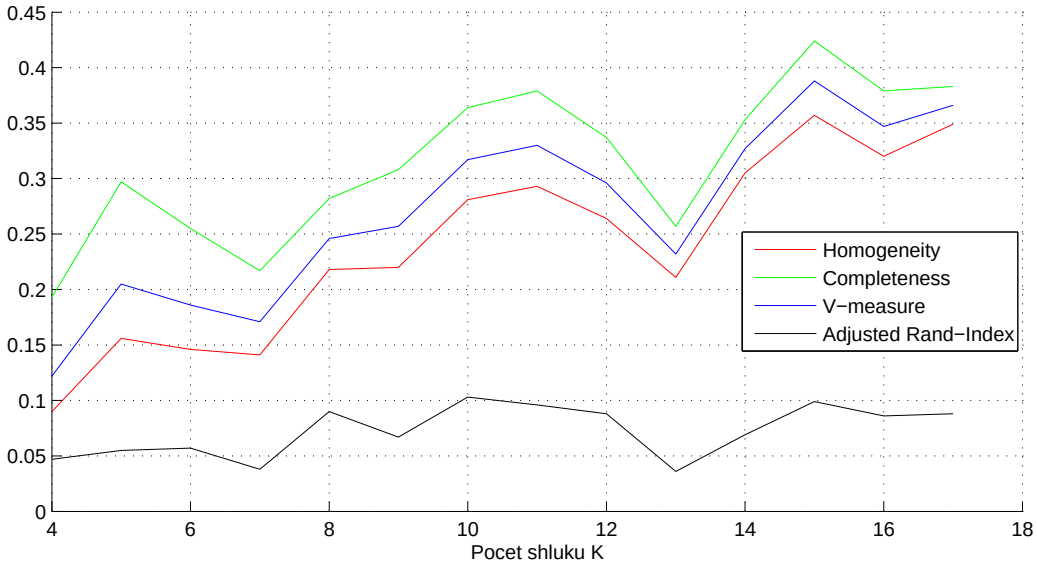
4.3.1 Hashing Vectorizer + K-means

I v třetím experimentu postupujeme stejným způsobem, jenom hledáme nastavení pro obě verze dokumentů. Jak je již uvedeno v předchozím odstavci, nastavení je testováno vždy na neupravených článcích. První test určil, na kolik dimenzí bude redukován prostor při použití Hashing Vectorizeru.

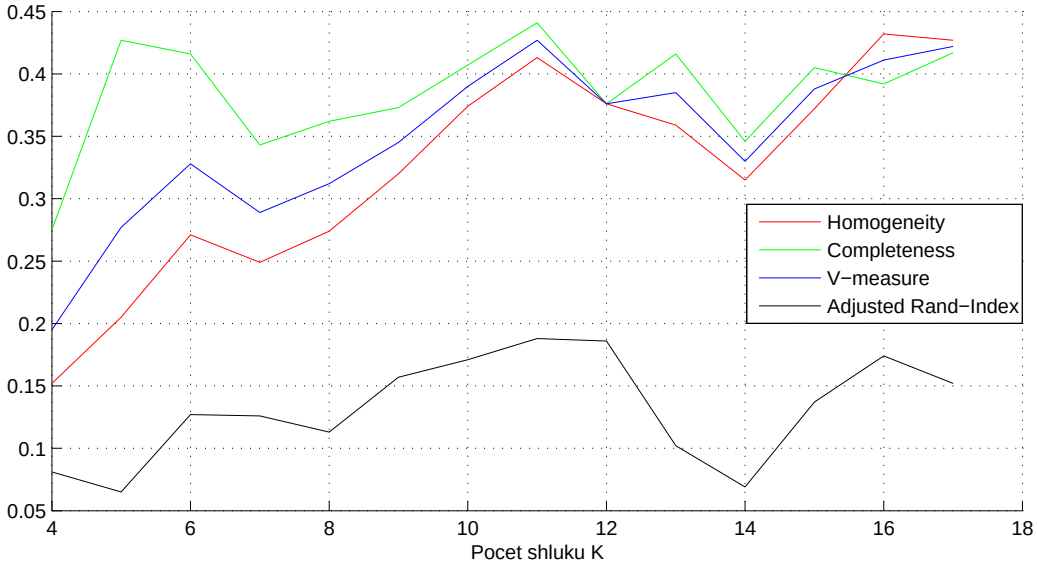


Obrázek 4.21: 3. Experiment (neupravená v.) - HV + K-means - Test snížení D

Test nastavení bohužel nebyl příliš průkazný a většina nastavení měla podobné výsledky. V dalších testech pro zjištění K bude tedy použito $D = 3200$, které mělo mírně lepší výsledek oproti ostatním.



Obrázek 4.22: 3. Experiment (neupravená v.) - HV + K-means - Test nastavení K



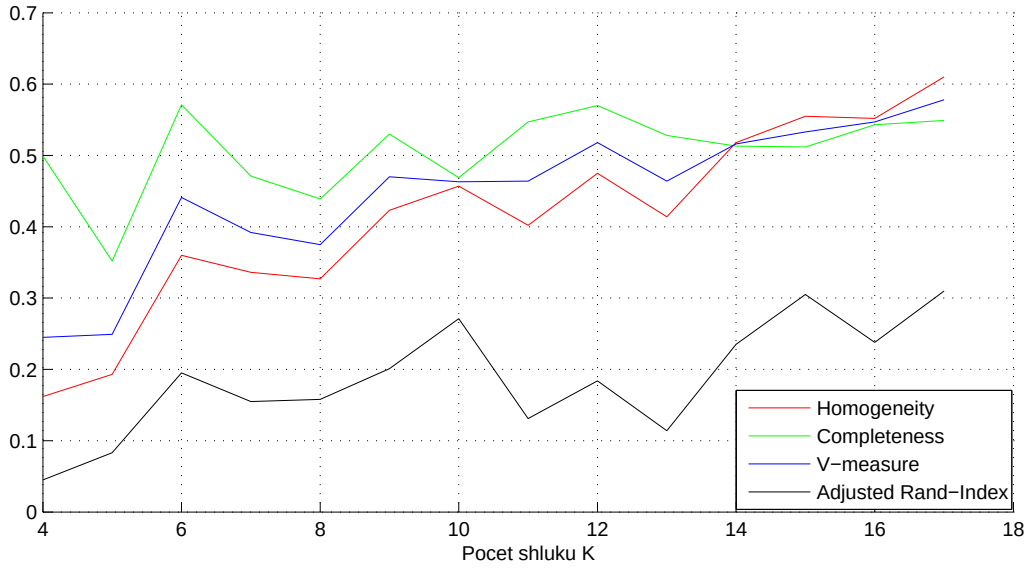
Obrázek 4.23: 3. Experiment (lemmatizovaná v.) - HV + K-means - Test nastavení K

Jednoznačný výsledek bohužel není ani u jedné z verzí partný, nicméně z grafu je jasně vidět jistý kvalitativní odskok u lemmatizované verze vůči verzi neupravené.

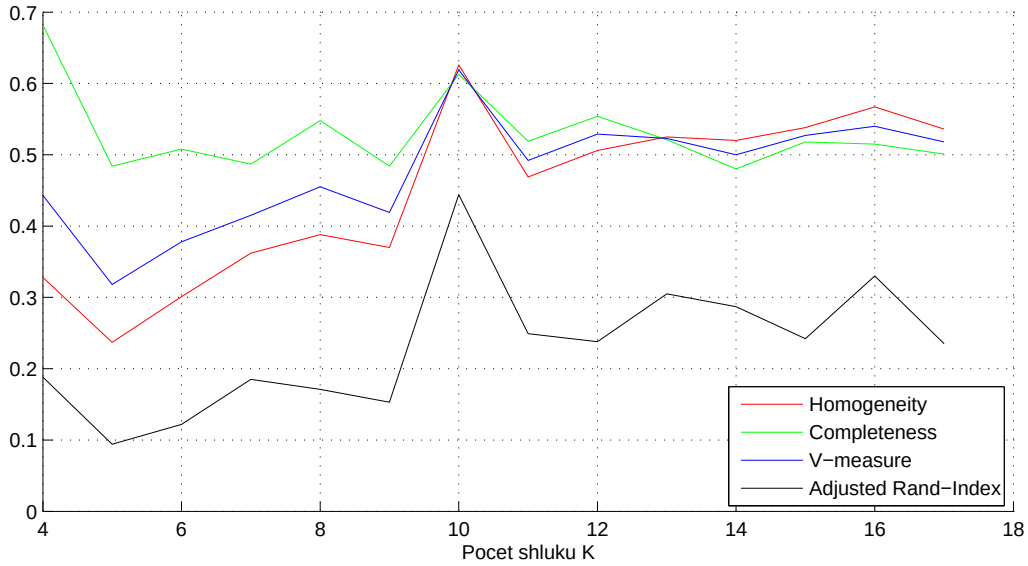
4.3.2 TF-IDF + K-means

Další pokus zkoumá výsledky TF-IDF transformace a K-means. Následující grafy ukazuje opět nastavení pro $K = 2, 3, ..., 17$. Jelikož u předchozího pokusu s Hashing Vectorizerem byl značný rozdíl mezi upravenou a neupravenou verzí, lze předpokládat, že kombinace TF-IDF a K-means, která měla v předchozích experimentech dobré výsledky, tento rozdíl nadále prohloubí. Zajímavé bude také sledovat, kolik díky lemmatizaci ubyde parametrů příznakového vektoru. V neupraveném textu totiž budou „vzdělaný“ a „vzdělaná“ představovat dva různé znaky, zatímco v lematizované verzi už jen jeden. Tím by mělo zaniknout velké množství šumu. Navíc TF-IDF transformace pracuje na principu počtu výskytů slov v dokumentech [2], a proto doslova vyžaduje podobnou úpravu vstupu.

[2] Jak již bylo řečeno, principem TF-IDF je předpoklad, že méněkrát použitá slova mají pro dokument větší význam



Obrázek 4.24: 3. Experiment (neupravená v.) - TF-IDF + K-means - Test nastavení K

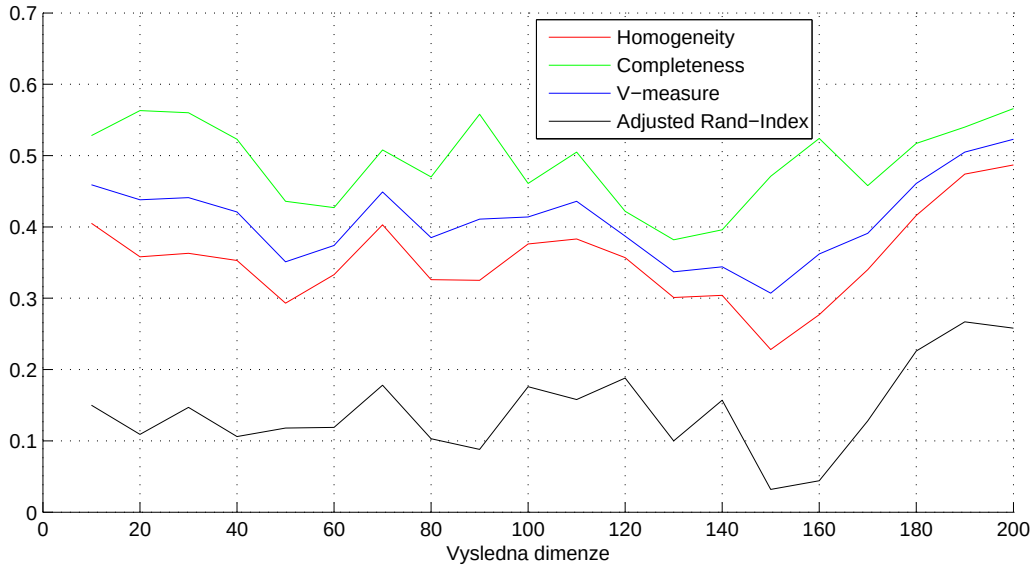


Obrázek 4.25: 3. Experiment (lemmatiz. v.) - TF-IDF + K-means - Test nastavení K

Z tohoto pokusu je podle předpokladu vidět značný rozdíl mezi lemmatizovanými a nelemmatizovanými dokumenty. Také počet dimenzí u lemmatizovaných dokumentů je výrazně nižší, pouze 4351 slov oproti 6286. To znamená, že u první verze tvoří šum téměř dva tisíce slov.

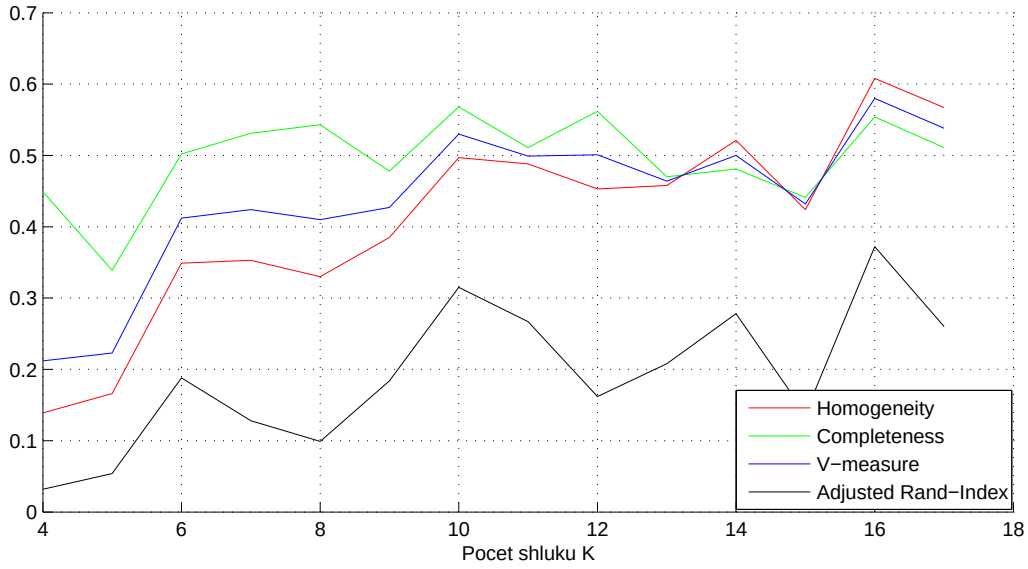
4.3.3 TF-IDF + PCA + K-means

Prvním krokem je znovu hledání ideálního nastavení snížení dimenze.

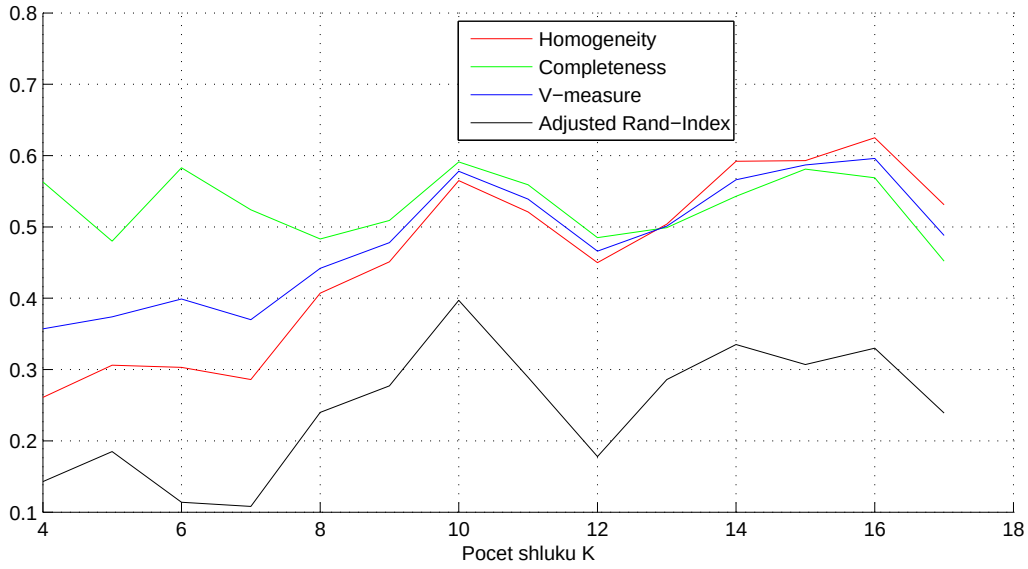


Obrázek 4.26: 3. Experiment (neupravená v.) - TF-IDF + PCA + K-means - Test nastavení dimenze

Zdá se, že nejlepším nastavením je ponechat maximální možné množství $D = 200$, neboť oblast $D > 180$ má výrazně lepší výsledky než ostatní nastavení. Proto oba dva testy pro porovnání budou vypočeteny s tímto nastavením.



Obrázek 4.27: 3. Experiment (neupravená v.) - TF-IDF + PCA + K-means - Test nastavení K

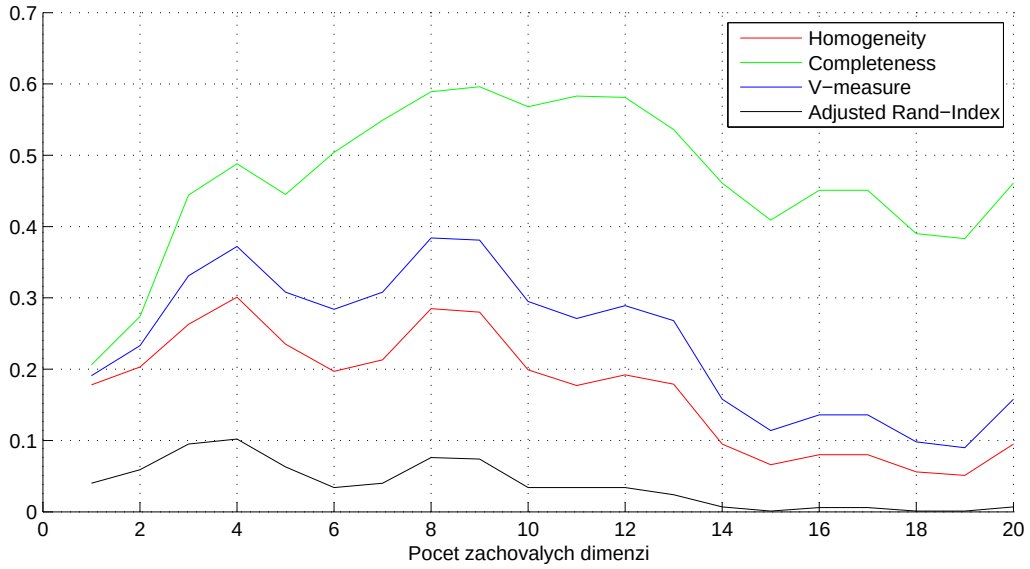


Obrázek 4.28: 3. Experiment (lemmatiz. v.) - TF-IDF + PCA + K-means - Test nastavení K

I v pokusu s kombinací algoritmů PCA, K-means a TF-IDF transformace výsledky potvrdily předpoklady. Lemmatizovaná verze i v tomto testu ukázala, že shluková analýza na takto upravených datech dosáhne lepších výsledků, ačkoliv tento skok nebyl tak markantní, jako u předchozího pokusu. Z toho lze usoudit, že PCA sama o sobě odstraňuje část šumu, ovšem část informace ztrácí.

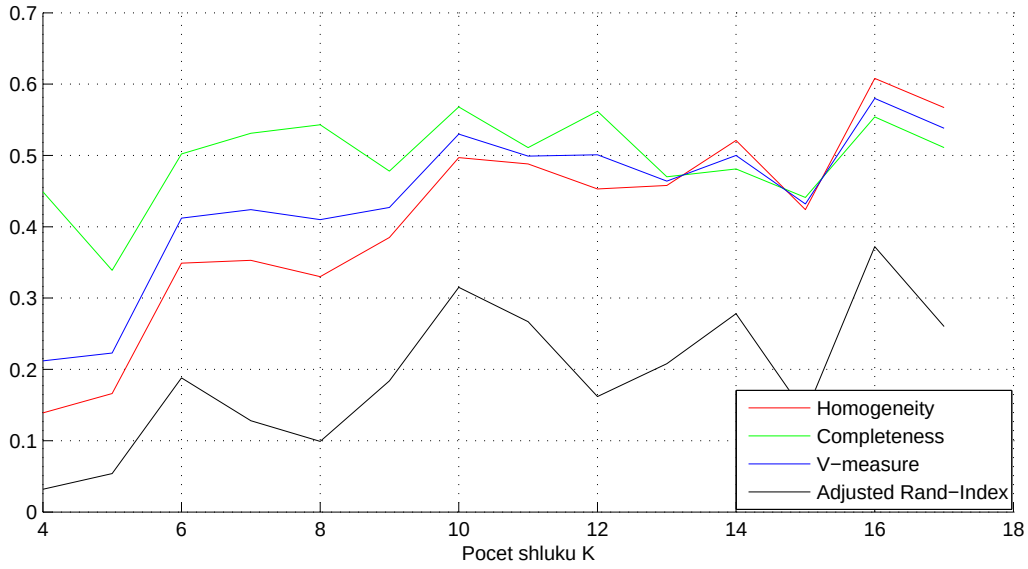
4.3.4 LSA + K-means

U rozsáhlejších souborů se pouze odhaduje, nakolik dimenzi snížit, respektive kolik vektorů by mělo zůstat pro další práci, v našem případě shlukování. V závislosti na velikosti souboru vybere přiměřený počet dimenzí (obvykle $\frac{1}{5}$ až $\frac{1}{2}$). Přestože výpočet byl časově náročnější než v předchozích experimentech, pokusil jsem se v následujícím testu najít vhodné nastavení.

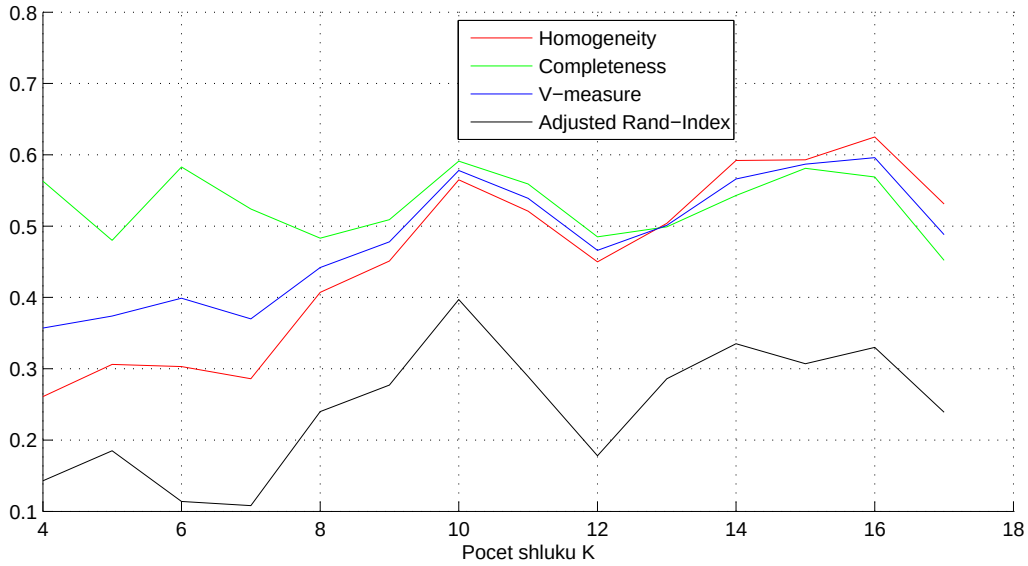


Obrázek 4.29: 3. Experiment (neupravená v.) - LSA + K-means - Test nastavení dimenze

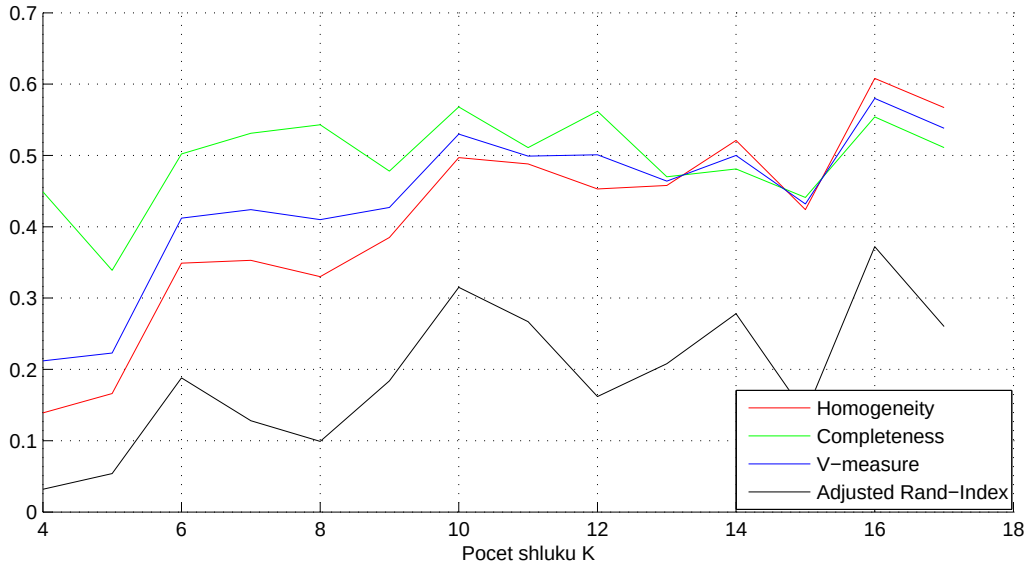
Protože pro žádná nastavení nebyla jednoznačně vhodná pro dobré shluknutí, další test proběhl pro $D = 4$, které mělo nejlepší výsledky a pro $D = 40$, které zahrnuje první pětinu z celkového počtu dimenzí.



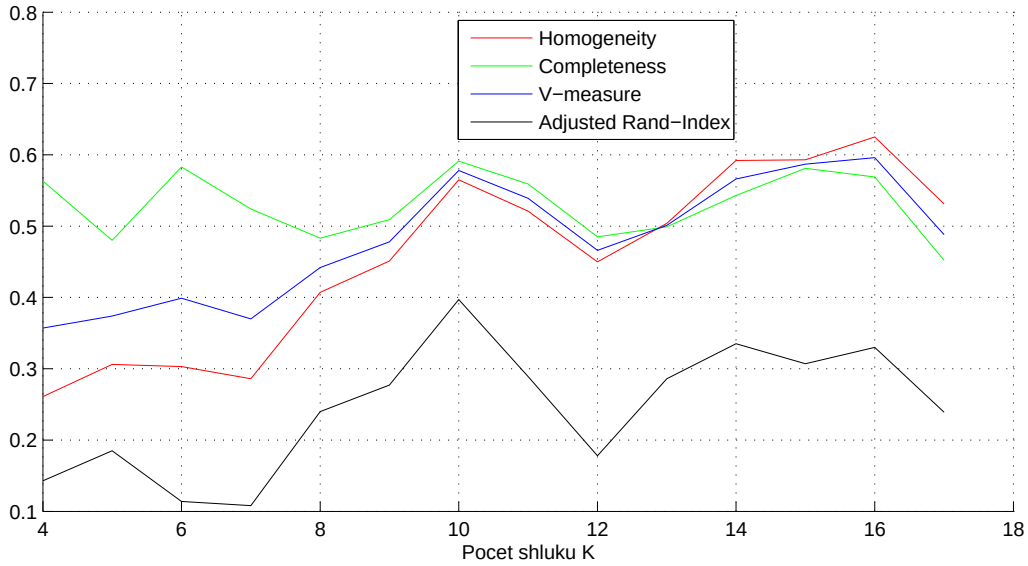
Obrázek 4.30: 3. Experiment (neupravená v.) - LSA + K-means - Test nastavení K pro D=4



Obrázek 4.31: 3. Experiment (lemmatiz. v.) - LSA + K-means - Test nastavení K pro D=4



Obrázek 4.32: 3. Experiment (neupravená v.) - LSA + K-means - Test nastavení K pro D=40



Obrázek 4.33: 3. Experiment (lemmatiz. v.) - LSA + K-means - Test nastavení K pro D=40

Ani jeden ze dvou testů bohužel neměl dobře interpretovatelné výsledky. Ačkoliv u lemmatizovaných článků je pozorovatelná mírně lepší kvalita analýzy, je v grafech dobře

pozorovatelný zajímavý jev. Nejlepší výsledky jsou většinou pro K vyšší než 10, ale nevykazují žádné skokové změny jako při testovaní předchozích metod. To znamená, že parametry obrazů byly tak nejasné, že shlukování nemělo valný význam, šlo spíše o náhodné rozdělení, přičemž velká většina článků (často až 90%, tedy 180 článků) skončila v jednom shluku.

4.3.5 Shrnutí

Ve všech testech včetně těch, kde výsledky byly téměř neprůkazné, byly tyto vždy lepší pro soubor dokumentů s lemmatizovaným textem. To bylo zvláště markantní u kombinací algoritmů, které využívají TF-IDF transformaci. Ta totiž přímo pracuje nejenom s počtem výskytů slov v jednotlivých článcích, ale také s počtem dokumentů slovo obsahujících. A protože přesně 1935 slov, tedy parametrů či dimenzí zaniklo lemmatizací (to znamená, že jejich výskyt byl spojen s nějakým příbuzným slovem), lze předpovědět daleko přesnější transformaci, a tudíž i lepší výsledky shluknutí. Výsledky se v některých případech zlepšily natolik, že šlo zcela jednoznačně na základě testování kvality analýzy vidět skok ve kvalitě pro deset shluků, což by měl být právě správný počet clusterů.

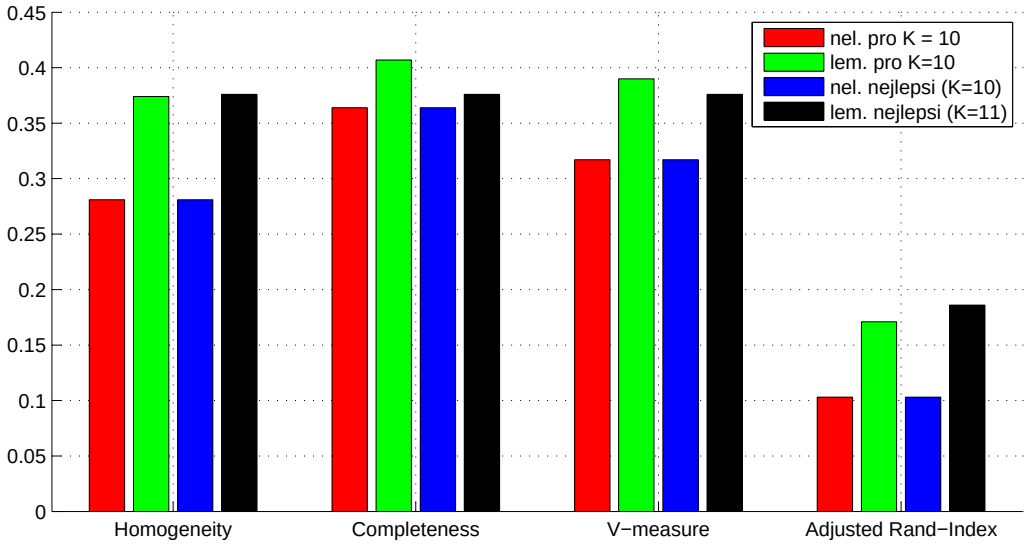
Nutno zmínit, že lemmatizace samozřejmně není dokonalá a nedokáže odstranit všechna úskalí českého textu. Například nerozezná, kdy bylo slovo „myslí“ použito jako sloveso a kdy jako podstatné jméno. Protože se jedná o lemmatizaci strojovou, v každém případě jej převede buďto na základní tvar podstatného jména, tedy „mysl“, nebo na základní tvar slovesa, tedy „myslet“.

V tomto experimentu se mi ani přes velké množství vyzkoušených kombinací nepodařilo najít vhodné nastavení pro latentní sémantickou analýzu, proto její výsledky nepřinášejí srozumitelná data. Jednalo se spíš o téměř náhodné shluky v závislosti na tom, jak pseudonáhodný start K-means nastavil počáteční podmínky a na cílovém počtu shluků.

Následující tabulky a grafy porovnávají vliv lematizace na vybrané znaky kvality shlukovací:

Tabulka 4.4: 3. Experiment - Srovnání kvality shlukovaní obou verzí - HV + K-means

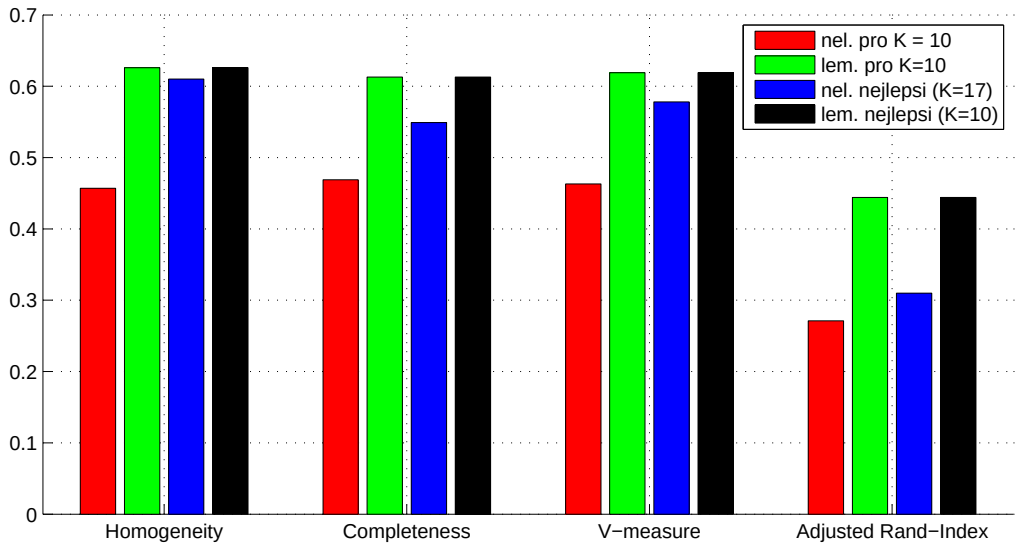
K-means HV	Homogeneity	Completeness	V-measure	Adj. Rand-Index
nel. K=10	0,281	0,364	0,317	0,103
lem. K=10	0,374	0,407	0,390	0,171
nej. nel. K=10	0,281	0,364	0,317	0,103
nej. lem. K=11	0,376	0,376	0,376	0,186



Obrázek 4.34: 3. Experiment - Srovnání kvality shlukovaní obou verzí - HV + K-means

Tabulka 4.5: 3. Experiment - Srovnání kvality shlukovaní obou verzí - TF-IDF + K-means

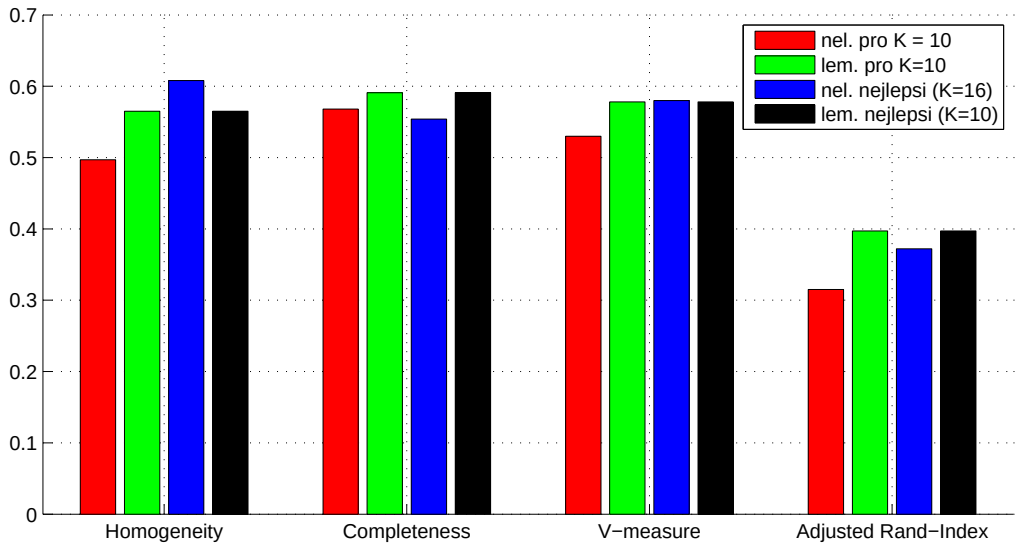
K-means HV	Homogeneity	Completeness	V-measure	Adj. Rand-Index
nel. K=10	0,457	0,469	0,463	0,271
lem. K=10	0,626	0,613	0,619	0,444
nej. nel. K=17	0,610	0,549	0,578	0,310
nej. lem. K=10	0,626	0,613	0,619	0,444



Obrázek 4.35: 3. Experiment - Srovnání kvality shlukovaní obou verzí - TF-IDF + K-means

Tabulka 4.6: 3. Experiment - Srovnání kvality shlukovaní obou verzí - PCA + K-means

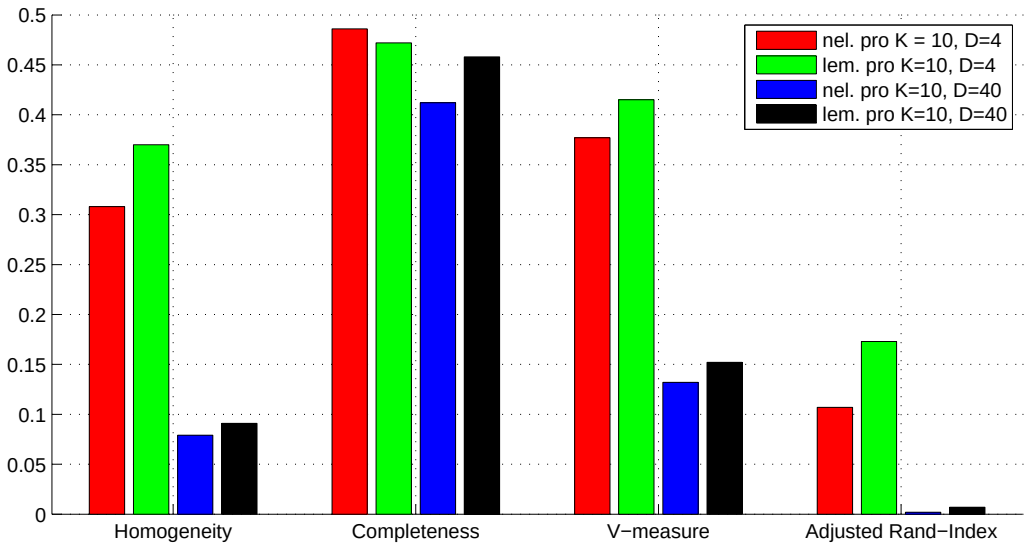
K-means HV	Homogeneity	Completeness	V-measure	Adj. Rand-Index
nel. K=10	0,497	0,568	0,530	0,315
lem. K=10	0,565	0,591	0,578	0,397
nej. nel. K=10	0,608	0,554	0,580	0,372
nej. lem. K=11	0,565	0,591	0,578	0,397



Obrázek 4.36: 3. Experiment - Srovnání kvality shlukovaní obou verzí - PCA + K-means

Tabulka 4.7: 3. Experiment - Srovnání kvality shlukovaní obou verzí - LSA + K-means

K-means HV	Homogeneity	Completeness	V-measure	Adj. Rand-Index
nel. K=10, D=4	0,308	0,486	0,377	0,107
lem. K=10, D=4	0,370	0,472	0,415	0,173
nel. K=10, D=40	0,079	0,412	0,132	0,002
lem. K=10, D=40	0,091	0,458	0,152	0,007



Obrázek 4.37: 3. Experiment - Srovnání kvality shlukovaní obou verzí - LSA + K-means

Kapitola 5

Závěr

Tato práce se zabývá zkoumáním teorie shlukovací analýzy zaměřené především na metody učení bez učitele vzhledem ke konkrétnímu problému, totiž rozdělení souboru novinových článků do skupin na základě jejich tématu.

Proběly tři pokusy. Cílem prvního pokusu bylo rozdělit 162 novinových článků v českém jazyce do devíti rozdílně velkých shluků. Tento pokus byl zacílen na vyzkoušení čtyř vybraných kombinací algoritmů a zjištění vhodnosti jejich aplikace na daný problém. Druhý experiment měl za účel otestovat správnost těchto kombinací na malém souboru 35 anglických článků rozdělených do sedmi stejně velkých shluků. Poslední z trojice pokusů se zabýval lemmatizací českého textu, čili převedením slov na základní tvar. Tato úprava článků měla za úkol zredukovat šum, který vzniká kvůli příbuzným slovům, ohýbání a dalším vlivům. Pokus byl proveden na dvou souborech o dvou stech článcích, které byly rozděleny do deseti shluků. První soubor obsahoval neupravené články, druhý články upravené lemmatizací.

V úvodu práce je uveden postup obvyklý při shlukové analýze od výběru znaků po interpretaci výsledků.

Pro **výběr znaků** byl v našem případě vybrán systém bag of words, tedy počet výskytu slov v článcích, které se dále upravovaly transformací Hashing Vectorizeru nebo TD-IDF. Protože jsme ve všech pokusech používali ke konečnému shluknutí algoritmus K-means,

mírou podobnosti byla euklidovská vzdálenost v N-rozměrném prostoru. Validací výsledků se zabývají následující odstavce.

Naším předpokladem bylo, že novinové články se společným tématem by měly tvořit přírodní a tím pádem kompaktní shluky. Mluvíme-li tedy o **shlukovacím kritériu**, lze jejich podobnost zaměnit taktéž s euklidovskou vzdáleností.

Vedoucí práce navrhla konkrétní **shlukovací algoritmy**, které by mohly být vhodné k řešení problému. První část mého úkolu bylo aplikovat tyto algoritmy pomocí modulu scikit-learn v programovacím jazyce Python. Protože jedna z metod, kterou jsem se rozhodl vyzkoušel v experimentální části práce, není součástí tohoto modulu, bylo nutné ji naprogramovat. Aby všechny experimenty bylo možné spustit z jednoho programu, rozhodl jsem se použít stejný programovací jazyk.

Jakmile proběly všechny experimenty, bylo třeba zpracovat všechna výstupní data vygenerovaná do 63 textových souborů a ověřit tak získané výsledky. Nejpřehlednější formou **validace výsledků** z výstupních dat se jevilo vykreslit je do přehledných a snadno čitelných grafů. K tomu jsem použil matematický software MATLAB, neboť s ním mám zkušenosti z předchozího studia a jeho možnosti k vyhodnocení dat se mi zdály vhodné. Všechny grafy si lze nechat znovu vykreslit spuštěním m-filu a načtením správného workspace, které obsahuje importovaná data k jednotlivým pokusům.

Inerpretací výsledků je porovnání vhodnosti jednotlivých algoritmů vzhledem k reálnému problému.

Ze čtyř testovaných kombinací algoritmů nejlepších výsledků dosahovala kombinace TF-IDF transformace a algoritmu K-means. Jen o trochu horších výsledků dosahovala kombinace TF-IDF, PCA a K-means, ale protože tato kombinace byla výpočtově méně náročná, byl pokus rychlejší a to především u malých souborů, kde byla rychlost větší až o jeden řád (2. experiment). Využití Hashing Vectorizeru a K-means bylo velice citlivé na nastavení správné dimenze. Tento krok by bez znalosti správného výsledku byl

velice obtížný. Výrazně horších a v některých případech i velmi nepřesných výsledků však celkem překvapivě dosahovala nejmodernější ze všech čtyř metod, latentní sémantická analýza. Příčinou je, že tato metoda není přímo určená k aplikaci, k jaké byla využita v mých experimentech. Zatímco předchozí tři metody jsou závislé na znalosti, popřípadě správném odhadu počtu shluků, do kterým mají být dokumenty roztříděné, nabízí LSA modernější přístup, který umožňuje porovnávat vztah mezi kterýmkoliv slovem vyskytujícím se v souboru dokumentů a dokumentem samotným. Pokud si parametry každého článku představíme jako rozměr N-dimenzionálního prostoru a článek do tohoto prostoru umístíme, dává nám LSA možnost do téhož prostoru umístit i všechna slova, která se v textu vyskytují a tím pádem sledovat jejich vzájemné relace.

U tří ze čtyř kombinací algoritmů se nejdříve muselo zjistit optimální nastavení. Konkrétně to byl Hashing Vectorizer, PCA a LSA. U všech jmenovaných nás zajímalo nastavení optimální redukce dimenze. Kvůli tomu u těchto metod proběhl jeden test, popřípadě více testů v případech, kde bylo nutné udělat test s hrubším a jemnějším nastavením, které měly pomoci vhodnou redukci najít.

Grafy zhodnocující každý experiment byly také vygenerovány v MATLABu. Obvykle jsem z každé metody vybral běh, který nastavením odpovídal takovému počtu shluků jako správný výsledek určený expertem. Pokud se tento běh lišil od toho s nejlepším hodnocením kvality shlukovací analýzy, byly do hodnocení vybrány oba. Díky porovnání těchto výsledků je možné odhadnout, zda metoda a její nastavení byly pro daný problém vhodné, či nikoliv. Pro bližší prozkoumání je ale nutné vrátit se ke grafům pro jednotlivá nastavení. Na některých z nich lze snadno vypozorovat vhodnost či nevhodnost algoritmu, porovnáním nastavení se správným počtem shluků určených expertem a okolí tohoto nastavení.

Z uvedeného vyplývá, že cíl této práce, tj. prozkoumat vybrané algoritmy a určit jejich vhodnost vzhledem k reálném problému, byl splněn.

Literatura

[1] Christopher D. Manning, Prabhakar Raghavan a Hinrich Schütze. *Introduction to Information Retrieval* Cambridge University Press. 2008.

[2] Sergios Theodoridis, Konstantinos Koutroumbas. *Pattern Recognition* Academic Press. 2008.

[3] Ka Yee Yeung, Walter L. Ruzzo. *Details of the Adjusted Rand index and Clustering algorithms Supplement to the paper „An empirical study on Principal Component Analysis for clustering gene expression data“ (to appear in Bioinformatics)* 2001.

[4] k-means clustering. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (Kalifornie): Wikimedia Foundation, 2001. Dostupné z: `http://en.wikipedia.org/wiki/K-means`

[5] Principal component analysis. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (Kalifornie): Wikimedia Foundation, 2001-. Dostupné z: `http://en.wikipedia.org/wiki/Principal_component_analysis`

[6] Latent semantic analysis. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (Kalifornie): Wikimedia Foundation, 2001-. Dostupné z: `http://en.wikipedia.org/wiki/Latent_semantic_analysis`

[7] Rand index. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (Kalifornie): Wikimedia Foundation, 2001-. Dostupné z: `http://en.wikipedia.org/wiki/Rand_index`

[8] tf–idf. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (Kalifornie): Wikimedia Foundation, 2001-. Dostupné z: `http://en.wikipedia.org/wiki/Tf%E2%80%93idf`

[9] Voronoi diagram. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (Kalifornie): Wikimedia Foundation, 2001-. Dostupné z: `http://en.wikipedia.org/wiki/Voronoi_diagram`

[10] Machine learning. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (Kalifornie): Wikimedia Foundation, 2001-. Dostupné z: `http://en.wikipedia.org/wiki/Machine_learning`

[11] Latent Semantic Analysis (LSA) Tutorial. In: *Personal Wiki* [online]. Gujarat (Indie), 2011. Dostupné z: `http://technowiki.wordpress.com/2011/08/27/latent-semantic-analysis-lsa-tutorial/`

[12] Clustering. In: *http://scikit-learn.org/* [online]. scikit-learn developers, 2010 - 2013. Dostupné z: `http://scikit-learn.org/stable/modules/clustering.html`

Kapitola 6

Přílohy

Tabulka 6.1: Tabulka hodnot k Obrázku 4.1

počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
100	0,310	0,333	0,321	0,171
200	0,419	0,447	0,433	0,245
300	0,395	0,434	0,414	0,219
400	0,417	0,544	0,472	0,175
500	0,397	0,492	0,440	0,187
600	0,421	0,475	0,446	0,246
700	0,369	0,572	0,449	0,141
800	0,384	0,478	0,426	0,168
900	0,472	0,543	0,505	0,289
1000	0,362	0,500	0,420	0,139
1100	0,442	0,566	0,497	0,216
1200	0,472	0,547	0,506	0,281
1300	0,390	0,507	0,441	0,163
1400	0,425	0,578	0,490	0,199
1500	0,463	0,557	0,505	0,273
1600	0,381	0,571	0,457	0,137
1700	0,469	0,553	0,507	0,274
1800	0,440	0,511	0,473	0,272
1900	0,472	0,518	0,494	0,264
2000	0,448	0,633	0,524	0,227
2100	0,416	0,463	0,438	0,209
2200	0,387	0,577	0,463	0,167
2300	0,426	0,490	0,456	0,207
2400	0,437	0,504	0,468	0,245
2500	0,405	0,478	0,439	0,232
2600	0,444	0,601	0,511	0,226
2700	0,433	0,598	0,503	0,204
2800	0,391	0,563	0,461	0,166
2900	0,421	0,530	0,470	0,199
3000	0,431	0,551	0,484	0,238

Tabulka 6.2: Tabulka hodnot k Obrázku 4.2

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
2	0,155	0,532	0,240	0,097
3	0,271	0,637	0,380	0,162
4	0,344	0,669	0,454	0,187
5	0,404	0,665	0,503	0,222
6	0,409	0,572	0,477	0,265
7	0,458	0,554	0,501	0,304
8	0,403	0,575	0,474	0,201
9	0,449	0,499	0,473	0,270
10	0,489	0,519	0,504	0,294
11	0,482	0,534	0,507	0,297
12	0,560	0,529	0,544	0,336
13	0,461	0,513	0,485	0,239
14	0,463	0,453	0,458	0,226
15	0,461	0,499	0,479	0,189
16	0,531	0,489	0,509	0,230

Tabulka 6.3: Tabulka hodnot k Obrázku 4.3

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
2	0,116	0,594	0,194	0,043
3	0,252	0,748	0,377	0,107
4	0,379	0,822	0,519	0,182
5	0,378	0,749	0,502	0,177
6	0,484	0,667	0,561	0,286
7	0,477	0,691	0,565	0,253
8	0,531	0,635	0,579	0,344
9	0,526	0,637	0,576	0,312
10	0,541	0,594	0,566	0,310
11	0,610	0,626	0,618	0,416
12	0,637	0,644	0,641	0,461
13	0,577	0,601	0,589	0,297
14	0,523	0,567	0,544	0,210
15	0,615	0,588	0,601	0,352
16	0,672	0,588	0,627	0,430

Tabulka 6.4: Tabulka hodnot k Obrázku 4.4

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
9	0,507	0,646	0,568	0,225
18	0,493	0,646	0,559	0,252
27	0,462	0,586	0,517	0,243
36	0,526	0,596	0,559	0,305
45	0,493	0,598	0,540	0,253
54	0,502	0,716	0,590	0,246
63	0,446	0,658	0,532	0,189
72	0,535	0,711	0,611	0,293
81	0,469	0,615	0,532	0,249
90	0,482	0,590	0,530	0,266
99	0,435	0,632	0,516	0,171
108	0,448	0,637	0,526	0,177
117	0,450	0,653	0,533	0,213
126	0,465	0,672	0,550	0,209
135	0,530	0,663	0,589	0,263
144	0,573	0,665	0,616	0,351
153	0,552	0,636	0,591	0,319
162	0,563	0,615	0,588	0,392

Tabulka 6.5: Tabulka hodnot k Obrázku 4.6

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
2	0,123	0,778	0,213	0,044
3	0,317	0,787	0,452	0,177
4	0,376	0,806	0,512	0,179
5	0,447	0,761	0,563	0,248
6	0,479	0,706	0,571	0,286
7	0,452	0,635	0,528	0,236
8	0,447	0,550	0,493	0,259
9	0,578	0,626	0,601	0,399
10	0,465	0,623	0,532	0,203
11	0,588	0,604	0,596	0,342
12	0,519	0,633	0,570	0,257
13	0,504	0,573	0,536	0,242
14	0,580	0,614	0,596	0,290
15	0,617	0,569	0,592	0,385
16	0,601	0,590	0,595	0,274

Tabulka 6.6: Tabulka hodnot k Obrázku 4.7

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
10	0,224	0,380	0,282	0,067
20	0,377	0,479	0,422	0,153
30	0,427	0,590	0,495	0,171
40	0,430	0,622	0,509	0,180
50	0,436	0,642	0,519	0,181
60	0,444	0,665	0,532	0,175
70	0,437	0,616	0,511	0,226
80	0,388	0,577	0,464	0,168
90	0,392	0,703	0,504	0,161
100	0,387	0,710	0,501	0,160
110	0,387	0,710	0,501	0,160
120	0,283	0,648	0,394	0,098
130	0,294	0,712	0,416	0,097
140	0,373	0,655	0,475	0,160
150	0,328	0,636	0,433	0,154
160	0,368	0,694	0,481	0,142

Tabulka 6.7: Tabulka hodnot k Obrázku 4.8

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
6	0,432	0,635	0,515	0,170
7	0,429	0,571	0,490	0,206
8	0,422	0,598	0,495	0,213
9	0,398	0,594	0,477	0,173
10	0,384	0,694	0,494	0,153
11	0,387	0,710	0,501	0,160
12	0,387	0,710	0,501	0,160
13	0,387	0,710	0,501	0,160
14	0,278	0,682	0,395	0,089
15	0,283	0,648	0,394	0,099
16	0,280	0,666	0,394	0,090
17	0,295	0,664	0,408	0,104
18	0,196	0,618	0,297	0,053
19	0,202	0,701	0,314	0,059
20	0,270	0,577	0,368	0,102
21	0,141	0,548	0,225	0,028
22	0,162	0,541	0,249	0,034
23	0,284	0,691	0,402	0,100
24	0,344	0,588	0,434	0,145
25	0,152	0,521	0,235	0,029

Tabulka 6.8: Tabulka hodnot k Obrázku 4.9

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
61	0,141	0,534	0,223	0,028
62	0,069	0,372	0,116	0,002
63	0,152	0,520	0,235	0,029
64	0,075	0,381	0,126	0,003
65	0,058	0,386	0,100	0,002
66	0,245	0,488	0,326	0,079
67	0,056	0,377	0,098	0,000
68	0,106	0,434	0,170	0,013
69	0,068	0,368	0,115	0,001
70	0,140	0,507	0,220	0,028
71	0,056	0,378	0,098	0,000
72	0,150	0,474	0,228	0,030
73	0,074	0,372	0,123	0,001
74	0,155	0,512	0,238	0,034
75	0,074	0,375	0,124	0,001
76	0,067	0,365	0,114	0,001
77	0,057	0,380	0,099	0,001
78	0,136	0,492	0,213	0,027
79	0,052	0,349	0,091	0,000
80	0,057	0,379	0,098	0,001

Tabulka 6.9: Tabulka hodnot k Obrázku 4.10

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
2	0,159	1,000	0,274	0,057
3	0,024	0,338	0,044	0,000
4	0,030	0,343	0,055	0,000
5	0,186	0,766	0,300	0,059
6	0,287	0,774	0,419	0,103
7	0,368	0,723	0,488	0,152
8	0,358	0,687	0,470	0,142
9	0,430	0,599	0,500	0,225
10	0,443	0,586	0,505	0,207
11	0,516	0,601	0,555	0,271
12	0,505	0,568	0,534	0,231
13	0,526	0,568	0,546	0,256
14	0,571	0,578	0,574	0,310
15	0,587	0,572	0,579	0,306
16	0,598	0,568	0,583	0,293

Tabulka 6.10: Tabulka hodnot k Obrázku 4.13

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
50	0,573	0,634	0,602	0,328
100	0,624	0,683	0,652	0,407
150	0,764	0,802	0,782	0,588
200	0,669	0,738	0,702	0,440
250	0,783	0,802	0,792	0,614
300	0,793	0,829	0,811	0,628
350	0,751	0,793	0,771	0,573
400	0,672	0,720	0,695	0,467
450	0,742	0,819	0,779	0,554
500	0,716	0,748	0,731	0,538
550	0,870	0,910	0,889	0,741
600	0,822	0,885	0,852	0,681
650	0,836	0,857	0,847	0,722
700	0,861	0,897	0,879	0,723
750	0,748	0,803	0,774	0,573
800	0,904	0,913	0,909	0,817
850	1,000	1,000	1,000	1,000
900	0,739	0,763	0,751	0,533
950	0,793	0,843	0,817	0,656
1000	0,809	0,832	0,820	0,655

Tabulka 6.11: Tabulka hodnot k Obrázku 4.14

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
2	0,283	0,885	0,429	0,167
3	0,364	0,747	0,489	0,214
4	0,498	0,739	0,595	0,329
5	0,716	0,911	0,802	0,571
6	0,843	0,924	0,881	0,767
7	0,960	0,963	0,962	0,928
8	0,883	0,892	0,887	0,762
9	0,878	0,833	0,855	0,721
10	0,963	0,858	0,908	0,805
11	1,000	0,856	0,922	0,826
12	0,960	0,796	0,871	0,700

Tabulka 6.12: Tabulka hodnot k Obrázku 4.15

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
2	0,211	1,000	0,348	0,086
3	0,409	1,000	0,581	0,206
4	0,612	0,943	0,743	0,443
5	0,796	1,000	0,887	0,691
6	0,898	1,000	0,946	0,825
7	1,000	1,000	1,000	1,000
8	1,000	0,953	0,976	0,950
9	1,000	0,910	0,953	0,895
10	1,000	0,871	0,931	0,836
11	1,000	0,835	0,910	0,772
12	1,000	0,816	0,899	0,749

Tabulka 6.13: Tabulka hodnot k Obrázku 4.16

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
1	0,610	0,692	0,649	0,310
3	0,870	0,910	0,889	0,741
5	0,960	0,963	0,962	0,928
7	1,000	1,000	1,000	1,000
9	1,000	1,000	1,000	1,000
11	1,000	1,000	1,000	1,000
13	1,000	1,000	1,000	1,000
15	0,960	0,963	0,962	0,928
17	0,921	0,930	0,925	0,865
19	0,904	0,913	0,909	0,817
21	0,891	0,893	0,892	0,815
23	0,960	0,963	0,962	0,928
25	0,960	0,963	0,962	0,928
27	0,960	0,963	0,962	0,928
29	0,960	0,963	0,962	0,928
31	0,939	0,950	0,944	0,874
33	0,960	0,963	0,962	0,928
35	1,000	1,000	1,000	1,000

Tabulka 6.14: Tabulka hodnot k Obrázku 4.17

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
2	0,211	1,000	0,348	0,086
3	0,458	0,903	0,607	0,302
4	0,593	1,000	0,744	0,370
5	0,796	1,000	0,887	0,691
6	0,898	1,000	0,946	0,825
7	1,000	1,000	1,000	1,000
8	1,000	0,953	0,976	0,950
9	1,000	0,910	0,953	0,895
10	1,000	0,871	0,931	0,836
11	1,000	0,835	0,910	0,772
12	1,000	0,802	0,890	0,702

Tabulka 6.15: Tabulka hodnot k Obrázku 4.18

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
1	0,374	0,475	0,419	0,121
3	0,465	0,694	0,557	0,148
5	0,758	0,898	0,822	0,509
7	0,775	0,900	0,833	0,547
9	0,807	0,904	0,853	0,634
11	0,922	0,949	0,936	0,839
13	0,665	0,815	0,733	0,421
15	0,588	0,774	0,668	0,363
17	0,542	0,764	0,634	0,279
19	0,497	0,744	0,596	0,216
21	0,216	0,495	0,301	0,013
23	0,331	0,566	0,418	0,074
25	0,553	0,764	0,642	0,278
27	0,449	0,578	0,506	0,184
29	0,207	0,475	0,289	0,002
31	0,177	0,449	0,253	-0,004
33	0,185	0,471	0,266	0,006
35	0,311	0,539	0,395	0,052

Tabulka 6.16: Tabulka hodnot k Obrázku 4.19

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
2	0,063	0,561	0,114	0,007
3	0,265	0,774	0,395	0,107
4	0,458	0,822	0,588	0,264
5	0,593	0,923	0,722	0,332
6	0,509	0,805	0,624	0,234
7	0,898	0,948	0,922	0,775
8	0,699	0,803	0,747	0,455
9	0,909	0,888	0,899	0,753
10	0,919	0,859	0,888	0,741
11	0,886	0,813	0,847	0,682
12	0,967	0,826	0,891	0,792
13	1,000	0,811	0,896	0,761
14	1,000	0,798	0,887	0,749
15	1,000	0,777	0,875	0,714
16	1,000	0,761	0,864	0,689
17	1,000	0,749	0,856	0,677

Tabulka 6.17: Tabulka hodnot k Obrázku 4.21

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
200	0,223	0,266	0,243	0,065
400	0,201	0,225	0,212	0,058
600	0,242	0,298	0,267	0,077
800	0,200	0,244	0,220	0,064
1000	0,278	0,363	0,315	0,093
1200	0,232	0,301	0,262	0,073
1400	0,231	0,306	0,263	0,061
1600	0,236	0,271	0,253	0,102
1800	0,194	0,269	0,225	0,044
2000	0,220	0,314	0,259	0,073
2200	0,240	0,346	0,283	0,075
2400	0,231	0,276	0,252	0,075
2600	0,199	0,261	0,226	0,041
2800	0,220	0,278	0,245	0,057
3000	0,267	0,348	0,302	0,086
3200	0,321	0,383	0,349	0,146
3400	0,221	0,296	0,253	0,065
3600	0,264	0,322	0,290	0,113
3800	0,292	0,386	0,333	0,119
4000	0,248	0,303	0,273	0,081
4200	0,217	0,336	0,264	0,053
4400	0,259	0,341	0,294	0,079
4600	0,234	0,343	0,278	0,057
4800	0,196	0,269	0,226	0,042
5000	0,214	0,401	0,279	0,049
5200	0,266	0,316	0,289	0,086
5400	0,297	0,367	0,328	0,106
5600	0,244	0,373	0,295	0,055
5800	0,238	0,308	0,269	0,069
6000	0,276	0,328	0,300	0,107

Tabulka 6.18: Tabulka hodnot k Obrázku 4.22

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,090	0,193	0,122	0,047
5	0,156	0,297	0,205	0,055
6	0,146	0,255	0,186	0,057
7	0,141	0,217	0,171	0,038
8	0,218	0,282	0,246	0,090
9	0,220	0,308	0,257	0,067
10	0,281	0,364	0,317	0,130
11	0,293	0,379	0,330	0,096
12	0,264	0,337	0,296	0,088
13	0,211	0,257	0,232	0,036
14	0,305	0,353	0,327	0,069
15	0,357	0,424	0,388	0,099
16	0,320	0,379	0,347	0,086
17	0,349	0,383	0,366	0,088

Tabulka 6.19: Tabulka hodnot k Obrázku 4.23

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,152	0,275	0,195	0,081
5	0,205	0,427	0,277	0,065
6	0,271	0,416	0,328	0,127
7	0,249	0,343	0,289	0,126
8	0,274	0,362	0,312	0,113
9	0,320	0,373	0,345	0,157
10	0,374	0,407	0,390	0,171
11	0,413	0,441	0,427	0,188
12	0,376	0,376	0,376	0,186
13	0,359	0,416	0,385	0,102
14	0,315	0,346	0,330	0,069
15	0,372	0,405	0,388	0,137
16	0,432	0,392	0,411	0,174
17	0,427	0,417	0,422	0,152

Tabulka 6.20: Tabulka hodnot k Obrázku 4.24

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,162	0,499	0,245	0,045
5	0,193	0,352	0,249	0,083
6	0,360	0,571	0,441	0,195
7	0,336	0,471	0,392	0,155
8	0,327	0,439	0,375	0,158
9	0,423	0,530	0,470	0,201
10	0,457	0,469	0,463	0,271
11	0,402	0,547	0,464	0,131
12	0,475	0,570	0,518	0,184
13	0,414	0,528	0,464	0,114
14	0,518	0,513	0,516	0,235
15	0,555	0,512	0,533	0,305
16	0,552	0,543	0,547	0,238
17	0,610	0,549	0,578	0,310

Tabulka 6.21: Tabulka hodnot k Obrázku 4.25

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,328	0,682	0,443	0,188
5	0,237	0,484	0,318	0,094
6	0,301	0,508	0,378	0,122
7	0,362	0,487	0,415	0,185
8	0,388	0,548	0,455	0,171
9	0,370	0,484	0,419	0,153
10	0,626	0,613	0,619	0,444
11	0,469	0,519	0,492	0,249
12	0,506	0,554	0,529	0,238
13	0,525	0,521	0,523	0,305
14	0,520	0,480	0,500	0,287
15	0,538	0,518	0,527	0,242
16	0,567	0,515	0,540	0,330
17	0,536	0,501	0,518	0,235

Tabulka 6.22: Tabulka hodnot k Obrázku 4.26

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
10	0,405	0,528	0,459	0,150
20	0,358	0,563	0,438	0,109
30	0,363	0,560	0,441	0,147
40	0,353	0,523	0,421	0,106
50	0,293	0,436	0,351	0,118
60	0,333	0,427	0,374	0,119
70	0,403	0,508	0,449	0,178
80	0,326	0,470	0,385	0,103
90	0,325	0,558	0,411	0,088
100	0,376	0,461	0,414	0,176
110	0,383	0,505	0,436	0,158
120	0,357	0,422	0,387	0,188
130	0,301	0,382	0,337	0,100
140	0,304	0,396	0,344	0,157
150	0,228	0,471	0,307	0,032
160	0,277	0,524	0,362	0,044
170	0,340	0,458	0,391	0,128
180	0,416	0,517	0,461	0,226
190	0,474	0,540	0,505	0,267
200	0,487	0,566	0,523	0,258

Tabulka 6.23: Tabulka hodnot k Obrázku 4.27

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,139	0,449	0,212	0,032
5	0,166	0,339	0,223	0,054
6	0,349	0,502	0,412	0,188
7	0,353	0,531	0,424	0,128
8	0,330	0,543	0,410	0,099
9	0,385	0,478	0,427	0,184
10	0,497	0,568	0,530	0,315
11	0,488	0,511	0,499	0,267
12	0,453	0,562	0,501	0,162
13	0,458	0,470	0,464	0,208
14	0,521	0,481	0,500	0,278
15	0,424	0,441	0,432	0,138
16	0,608	0,554	0,580	0,372
17	0,567	0,511	0,538	0,260

Tabulka 6.24: Tabulka hodnot k Obrázku 4.28

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,261	0,563	0,357	0,143
5	0,306	0,480	0,374	0,185
6	0,303	0,583	0,399	0,114
7	0,286	0,524	0,370	0,108
8	0,407	0,483	0,442	0,240
9	0,451	0,509	0,478	0,277
10	0,565	0,591	0,578	0,397
11	0,521	0,559	0,539	0,289
12	0,450	0,485	0,466	0,178
13	0,504	0,499	0,501	0,286
14	0,592	0,543	0,566	0,335
15	0,593	0,581	0,587	0,307
16	0,625	0,569	0,596	0,330
17	0,531	0,452	0,488	0,239

Tabulka 6.25: Tabulka hodnot k Obrázku 4.29

Počet dimenzí	Homogeneity	Completeness	V-measure	Adj. Rand-Index
1	0,178	0,206	0,191	0,040
2	0,203	0,274	0,233	0,059
3	0,263	0,444	0,331	0,095
4	0,301	0,488	0,372	0,102
5	0,235	0,445	0,308	0,063
6	0,197	0,504	0,284	0,034
7	0,213	0,549	0,308	0,040
8	0,285	0,589	0,384	0,076
9	0,280	0,596	0,381	0,074
10	0,199	0,568	0,295	0,034
11	0,177	0,583	0,271	0,034
12	0,192	0,581	0,289	0,034
13	0,179	0,536	0,268	0,024
14	0,095	0,461	0,158	0,007
15	0,066	0,409	0,114	0,001
16	0,080	0,451	0,136	0,006
17	0,080	0,451	0,136	0,006
18	0,056	0,390	0,098	0,001
19	0,051	0,383	0,090	0,001
20	0,095	0,461	0,158	0,007

Tabulka 6.26: Tabulka hodnot k Obrázku 4.30

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,106	0,689	0,183	0,028
5	0,115	0,652	0,196	0,028
6	0,130	0,596	0,214	0,029
7	0,185	0,485	0,268	0,048
8	0,194	0,473	0,275	0,044
9	0,226	0,499	0,311	0,048
10	0,308	0,486	0,377	0,107
11	0,304	0,482	0,373	0,104
12	0,300	0,447	0,359	0,097
13	0,298	0,421	0,349	0,079
14	0,365	0,439	0,399	0,151
15	0,421	0,460	0,439	0,186
16	0,422	0,454	0,438	0,177
17	0,458	0,461	0,459	0,197

Tabulka 6.27: Tabulka hodnot k Obrázku 4.31

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,145	0,605	0,234	0,036
5	0,145	0,616	0,235	0,036
6	0,293	0,599	0,393	0,153
7	0,288	0,475	0,359	0,140
8	0,267	0,433	0,330	0,099
9	0,319	0,463	0,378	0,132
10	0,370	0,472	0,415	0,173
11	0,390	0,471	0,427	0,198
12	0,425	0,487	0,454	0,209
13	0,439	0,486	0,461	0,211
14	0,441	0,481	0,460	0,214
15	0,434	0,465	0,449	0,202
16	0,438	0,440	0,439	0,192
17	0,458	0,456	0,457	0,198

Tabulka 6.28: Tabulka hodnot k Obrázku 4.32

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,015	0,369	0,029	0,000
5	0,025	0,389	0,048	0,000
6	0,027	0,345	0,050	0,000
7	0,060	0,435	0,105	0,004
8	0,058	0,400	0,102	0,001
9	0,042	0,353	0,076	0,000
10	0,079	0,412	0,132	0,002
11	0,085	0,445	0,143	0,006
12	0,073	0,382	0,123	0,001
13	0,073	0,374	0,122	0,000
14	0,079	0,364	0,130	0,001
15	0,088	0,381	0,144	0,001
16	0,124	0,415	0,192	0,004
17	0,166	0,365	0,228	0,013

Tabulka 6.29: Tabulka hodnot k Obrázku 4.33

Počet shluků K	Homogeneity	Completeness	V-measure	Adj. Rand-Index
4	0,020	0,397	0,039	0,000
5	0,041	0,452	0,076	0,002
6	0,132	0,695	0,223	0,033
7	0,053	0,412	0,094	0,002
8	0,067	0,442	0,117	0,003
9	0,084	0,428	0,140	0,003
10	0,091	0,458	0,152	0,007
11	0,103	0,451	0,167	0,008
12	0,104	0,423	0,167	0,003
13	0,268	0,537	0,357	0,074
14	0,098	0,395	0,157	0,001
15	0,174	0,460	0,253	0,011
16	0,222	0,413	0,289	0,060
17	0,316	0,497	0,386	0,097