Shluková analýza – specializované algoritmy

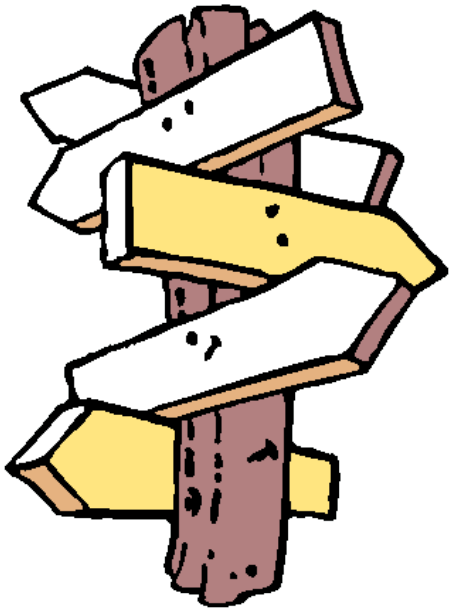
Jiří Kléma

Katedra kybernetiky,
FEL, ČVUT v Praze

http://ida.felk.cvut.cz

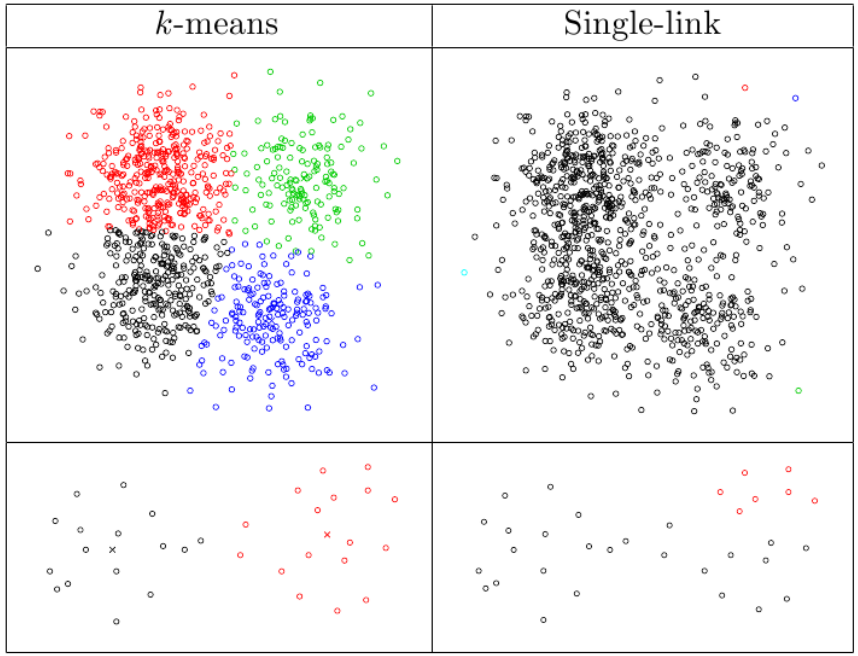
Struktura přednášky

- robustnost základních metod – příklady,
- složitost k-středů a hierarchického aglomerativního shlukování,
- hodnocení kvality rozkladu
  - interní versus externí,
- jak shlukování definovat?
- pokročilý přístup
  - spektrální shlukování,
- odvozené přístupy
  - konceptuální shlukování,
  - dvojshlukování,
  - učení s poloučitelem.

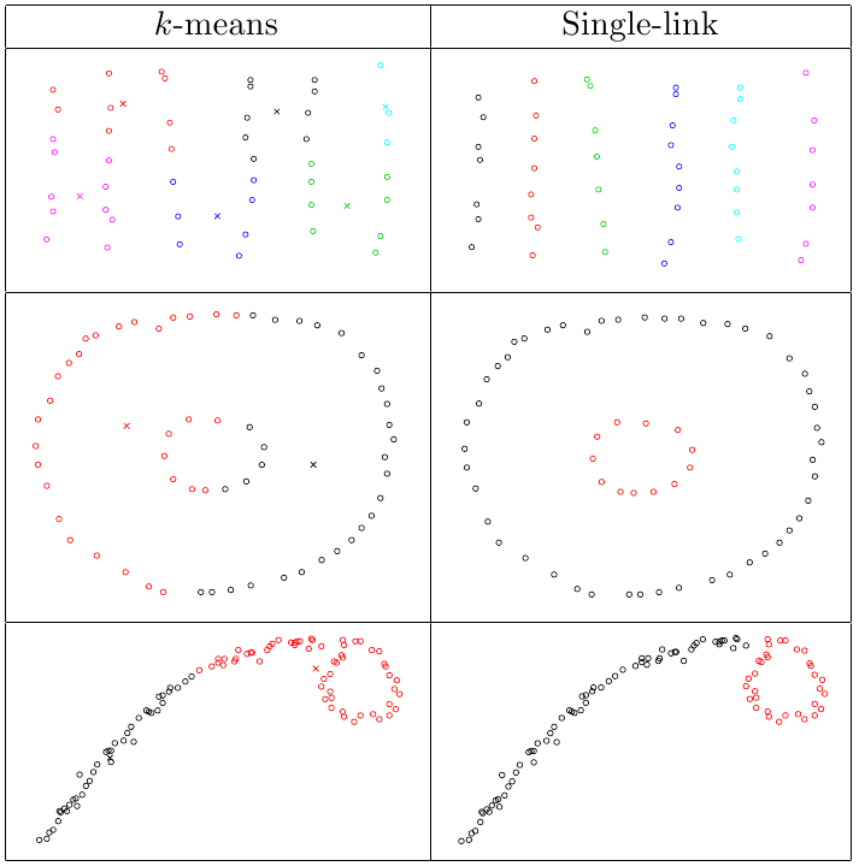


Srovnání k-středů a hierarchický single-link

- single linkage má tendenci k delším nekompaktním shlukům,
- k-středů tvoří kompaktní shluky, complete linkage je citlivá na odlehlé hodnoty,



k-středů intuitivně správně



single linkage intuitivně správně

Carnegie Mellon University, course: Statistics 36-350: Data Mining

Složitost – srovnání

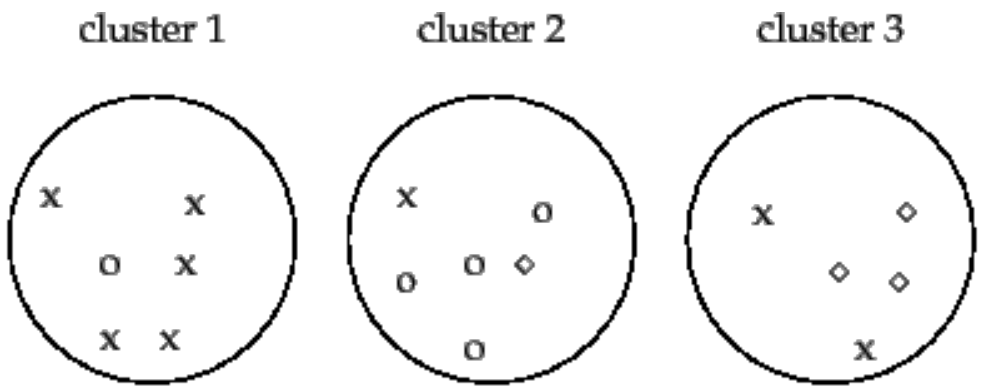
- předpoklad: $d(x_i, x_j) \in \mathcal{O}(n)$,
- algoritmus k-středů
  - přiřazení příkladů do shluků: $\mathcal{O}(km)$ výpočtů vzdálenosti $\rightarrow f_E \in \mathcal{O}(knm)$,
  - přepočtení středů: $f_M \in \mathcal{O}(nm)$ (každý příklad je použit jednou v některém z centroidů),
  - neznámý počet iterací do zastavení: i (odhady od konstantní s m až po $\mathcal{O}(m^{kn})$),
  - shrnutí: $f = i(f_E + f_M) \in \mathcal{O}(iknm)$,
- hierarchické aglomerativní shlukování (single link)
  - inicializace
    * napočítej vzdálenosti mezi všemi dvojicemi bodů $f_I \in \mathcal{O}(m^2 n)$,
    * *next-best-merge* pole – pro každý ze shluků nejbližší soused se vzdáleností (složitost skryta v předchozím),
  - $m - 1$ iterací k vytvoření úplného dendrogramu
    * najdi nejmenší vzdálenost v next-best-merge poli $f_{E_0} \in \mathcal{O}(m)$,
    * uprav matici vzdáleností $f_{E_1} \in \mathcal{O}(m)$,
    * uprav next-best-merge pole $f_{E_2} \in \mathcal{O}(m)$,
  - shrnutí: $f = f_I + (m-1)(f_{E_0} + f_{E_1} + f_{E_2}) \in \mathcal{O}(m^2 n)$.

Hodnocení kvality shlukování

- **interní:** vychází ze tří vlastností rozkladu
  - homogenita – jsou instance ve společných shlucích podobné?
    $hom = \frac{1}{m}\sum_{i=1}^{k}\sum_{x_j \in C_i} d(x_j, \mu_i)$,
  - separabilita – jsou instance z různých shluků nepodobné?
    $sep = \frac{1}{k(k-1)}\sum_{i=1}^{k}\sum_{j=1, j\neq i}^{k} \exp\left(-\frac{d^2(\mu_i,\mu_j)}{2\sigma^2}\right)$,
  - stabilita – kolik vztahů se zruší přidáním šumu nebo náhodným vzorkováním příkladů?
- základní interní hodnotící kritéria shlukovacích algoritmů
  - vnitroshluková variabilita: (viz homogenita u k-středů),
  - věrohodnost modelu: (pro pstní modely, viz EM GMM),
  - velikost vyváženého řezu grafu podobnosti: (viz spektrální shlukování později)
    $\frac{1}{2}\sum_{ij} s(x_i, x_j)(1 - \delta(C_i - C_j))(\frac{1}{|C_i|} + \frac{1}{|C_j|})$,
- shlukování je subjektivní, "objektivní" interní míra může být zavádějící.

Hodnocení kvality shlukování

- **externí:** srovnání rozkladu Ω se známou anotací $G = \{G_1, \ldots, G_c\}$ (gold standard),
- základní externí hodnotící kritéria
  - ryzost (purity)
    - celkový poměr příkladů z nejčastější třídy ve všech shlucích (každý shluk přiřazuje příklad do nejčastější třídy ve shluku), $purity(\Omega, G) = \frac{1}{m} \sum_{i=1\ldots k} \max_{j=1\ldots c} |C_i \cap G_j|$
    - nevýhoda: nelze použít pro srovnání rozkladů s různým k,
    - př. (obrázek): $ryzost = \frac{5+4+3}{17} = 0.71$



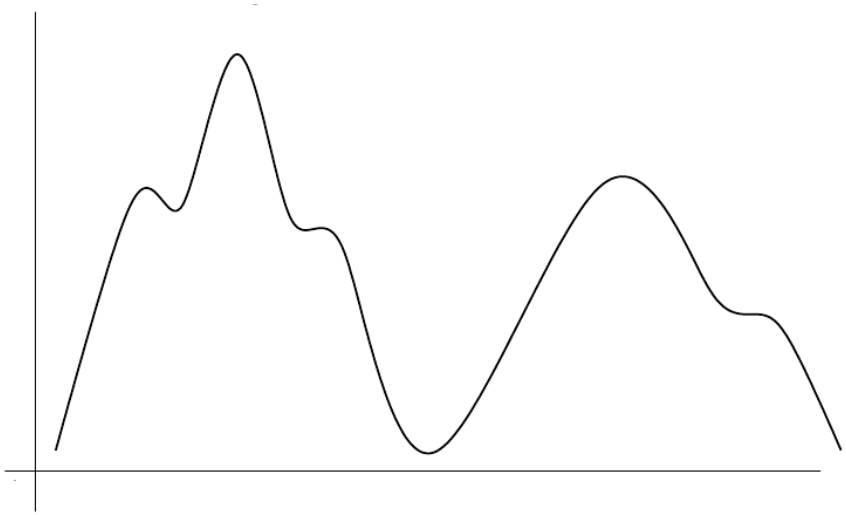
Manning et al.: Introduction to Information Retrieval.
http://nlp.stanford.edu/IR-book/html/htmledition/irbook.html

Hodnocení kvality shlukování

- základní externí hodnotící kritéria (pokračování)
  - normalizovaná vzájemná informace
    - vychází z informační entropie,
      $NMI(\Omega, G) = \frac{2I(\Omega,G)}{H(\Omega)H(G)}$
      $I(\Omega, G) = -\sum_{i=1\ldots k}\sum_{j=1\ldots c} P(C_i \cap G_j)\log_2\left(\frac{P(C_i \cap G_j)}{P(C_i)P(G_j)}\right)$
      $H(\Omega) = -\sum_{i=1\ldots k} P(C_i)\log_2(P(C_i))$, $H(G) = -\sum_{j=1\ldots c} P(G_j)\log_2(P(G_j))$
    - př. (obrázek): $H(\Omega) = -2\frac{6}{17}log_2(\frac{6}{17}) - \frac{5}{17}log_2(\frac{5}{17}) = 1.58$, $H(G) = 1.52$
      $NMI = \frac{2\times 0.96}{1.58\times 1.52} = 0.8$
  - rand index
    - shlukování interpretuje jako posloupnost $\frac{m(m-1)}{2}$ rozhodnutí,
    - rozhodnutí = rozklad buď zařadí dvojici příkladů do stejného shluku nebo ne,
    - $TP\ldots$ dvojice ve stejném shluku dle Ω a ve stejné třídě dle G,
    - $TN\ldots$ dvojice v různých shlucích dle Ω a v různých třídách dle G,
    - $RI(\Omega, G) = \frac{TP+TN}{TP+TN+FP+FN} = \frac{2(TP+TN)}{m(m-1)}$
    - př. (obrázek): $RI = \frac{2(\binom{5}{2}+\binom{4}{2}+\binom{3}{2}+\binom{2}{2}+5\times 8+1\times 7+4\times 5+1\times 2+1\times 3)}{17\times 16} = 0.68$

Jak shlukování definovat?

- kvalitativní funkcí – viz předchozí slajdy
  - \+ opora v matematice a optimalizaci,
  - – heuristická volba funkce, NP těžký problém,
- na základě hustoty
  - – shluk je oblast s velkou hustotou objektů, oblasti s málo objekty shluky oddělují,
  - \+ odhad pravděpodobnostní hustoty je známý problém,
  - – pro více dimenzí ještě složitější než shlukování,

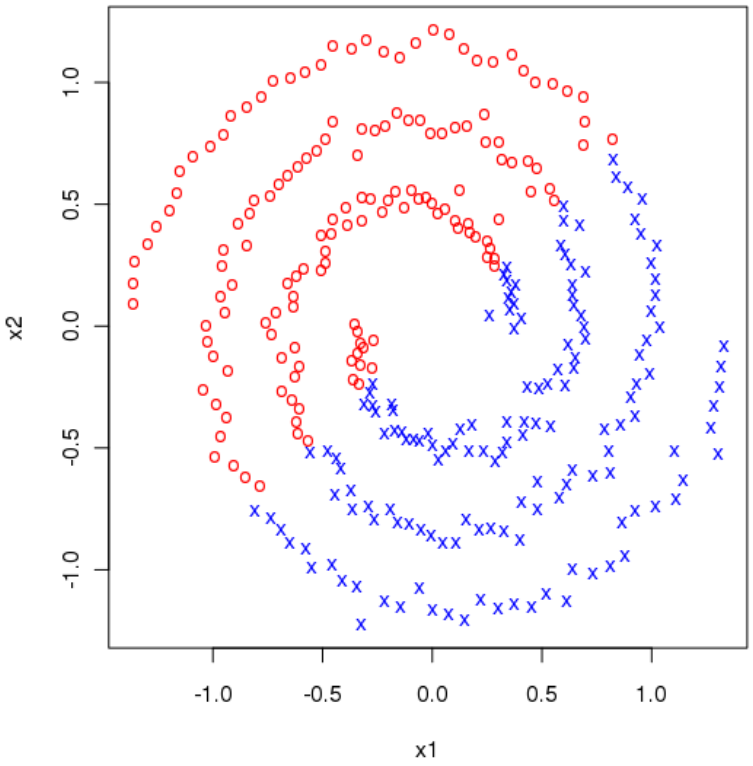


Jak shlukování definovat?

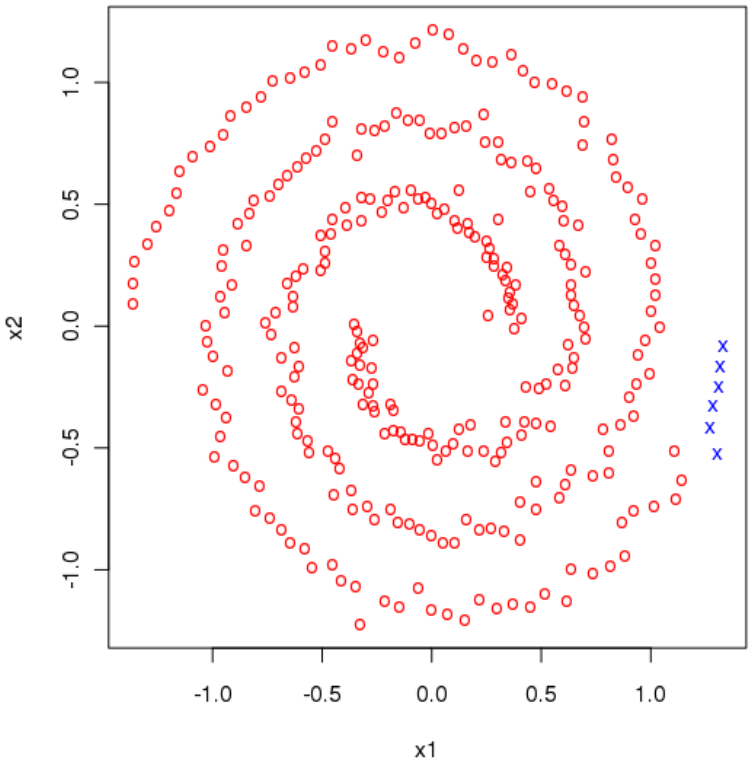
- generativním modelem
  - data byla vygenerována (pstním) modelem – ten i říká, co jsou shluky,
    - + model dává i výklad, nejenom rozklad, řada standardních postupů,
    - – příliš silné předpoklady s volbou modelu, odhad parametrů není snadný,
- axiomaticky
  - shlukovací fci od matic vzáleností k rozkladu nedefinuje přímo, ale na základě vlastností,
  - příklad systému axiomů:
    - ∗ invariance k měřítku: škálování vzdáleností nevede na změnu rozkladu,
    - ∗ bohatost: pro každý rozklad existuje nějaká matice vzdálenosti, která jej generuje,
    - ∗ konzistence: přibliž body uvnitř shluků, oddal ty mimo $\longrightarrow$ rozklad se nezmění,
    - + elegantní způsob definice,
    - – některé množiny axiomů sporné, malá užitečnost, spíše matematická hříčka,
- vyjít z teorie informace
  - shlukování=ztrátová komprese, definujeme přípustnou ztrátu či délku kódu,
    - – co je původní informace? opět nelze řešit analyticky.

Spektrální shlukování – motivace

- algoritmy shlukování předjímají shluky nějakých tvarů
  - potíže s nečekanými tvary (mj. lineárně neoddělitelnými shluky),
  - potíže s podobností založenou na klasické představě vzdálenosti.



Aplikace k-středů



Aplikace single linkage

R, kernlab package, specc function demo

Spektrální shlukování – souvislosti

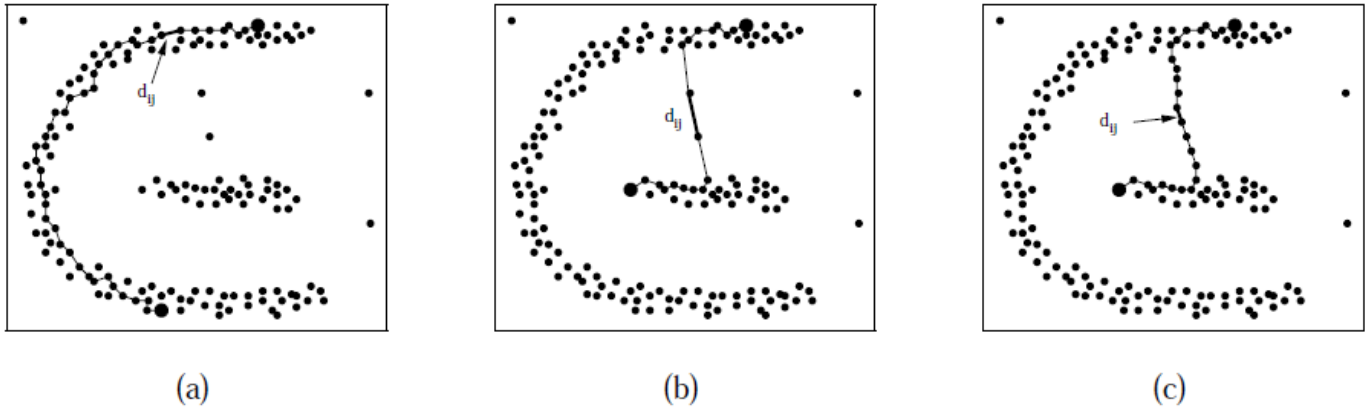
- řešením obvykle **transformace příznakového prostoru**,
- shlukovací algoritmus problémově nezávislý, transformace závisí na úloze
  - explicitní transformace
    - určení souřadnic objektů v novém příznakovém prostoru,
    - klasické shlukování v novém prostoru,
    - názorná, ale nepříliš praktická varianta,
  - implicitní transformace
    - prostřednictvím podobnostní resp. kernel funkce $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$,
    - pouze funkce dvojice objektů, nikoli souřadnice objektů v novém prostoru,
    - pro shlukování velmi přirozené, bez podobnosti se stejně neobejdeme,

    - analogie kernel triku obvyklého v klasifikaci (SVM),
    - možný přechod do prostoru vysoké dimenze, kde jsou shluky (třídy) dobře oddělitelné,

    - kernel PCA – aplikace PCA pro projekci do nízkodimenzionálního obrazu prostoru příznaků.

Příklad: spirály – kernel propojení, Gaussian kernel

- kernel propojení (connectivity kernel)
  - vzdálenost mezi dvěma body určena maximální délkou hrany na cestě mezi body,
  - ze všech cest volíme tu, která výše uvedené kritérium minimalizuje,
  - robustnější než single linkage,

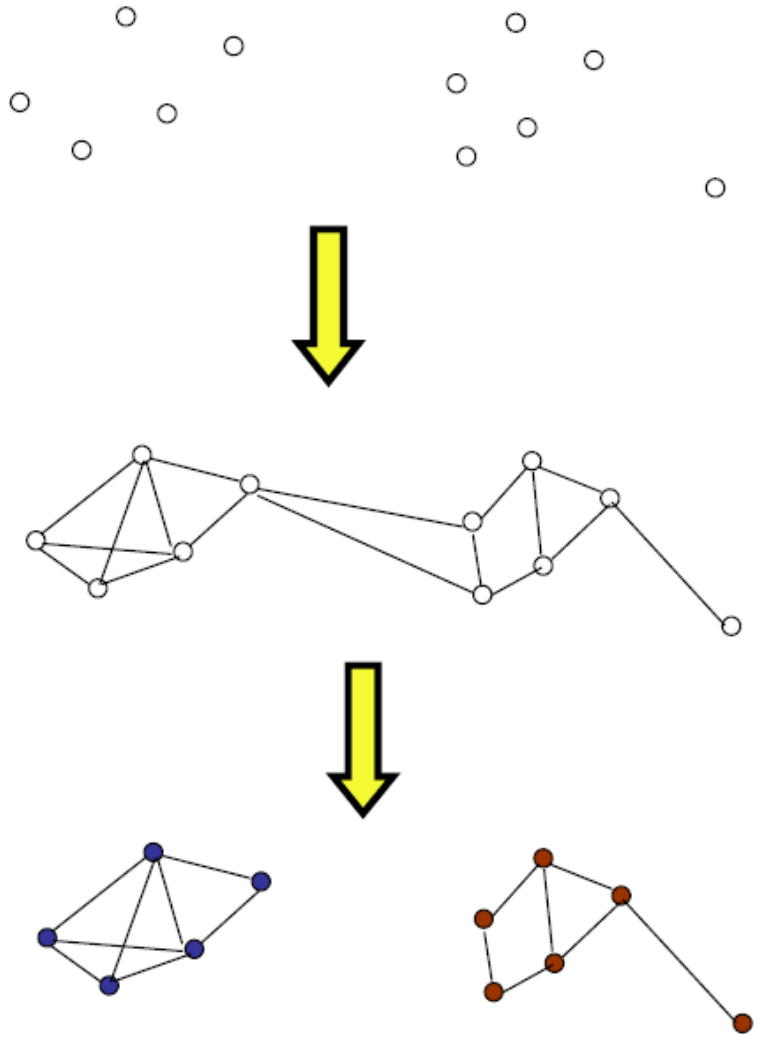


Fischer et al.: Clustering with the Connectivity Kernel

- Gaussian (RBF) kernel
  - $s(x_i, x_j) = \exp(-||x_i - x_j||/\sigma^2)$,
  - σ voleno tak, aby okolí bodu bylo "těsné",
  - implicitní příznakový prostor nekonečné dimenze.

Spektrální shlukování v kostce

- vstupem je množina objektů,

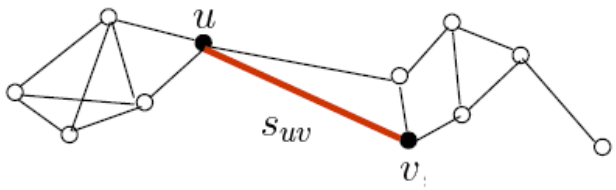


- popíšeme ji jako graf,
- hrany vyjadřují podobnost,
- graf rozložíme na komponenty = shluky,
- k rozkladu použijeme spektrální vlastnosti grafu.

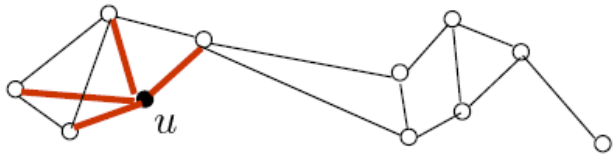
Azran: A Tutorial on Spectral Clustering

Teorie grafů – terminologie

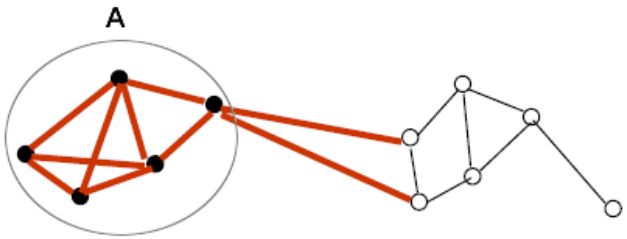
- podobnost dvojice uzlů (objektů)
  - $s_{uv} = \langle u, v \rangle$,



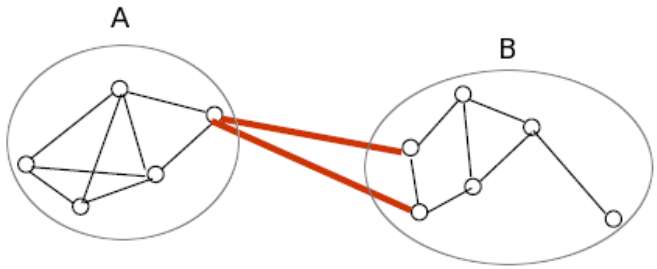
- stupeň uzlu, matice stupňů
  - $d_u = \sum_{v=1}^{m} s_{uv}$,
  - $\mathcal{D} = diag(d_1, \ldots, d_m)$,



- velikost a stupeň množiny vrcholů (shluku)
  - $|A|$ ... počet vrcholů v A,
  - $vol(A) = \sum_{u \in A} d_u$,



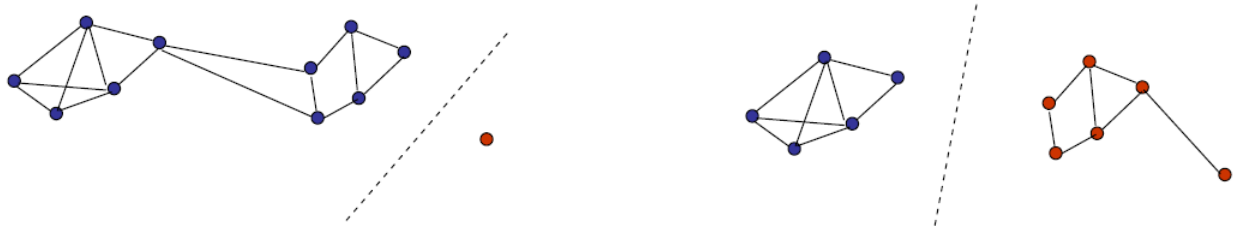
- hranový řez mezi dvěma komponentami
  - $cut(A, B) = \sum_{u \in A} \sum_{v \in B} s_{uv}$.



Azran: A Tutorial on Spectral Clustering

Spektrální shlukování jako zjednodušený minimální řez grafu

- shlukování $\sim$ rozklad grafu podobnosti na komponenty,
- lze řešit jako optimalizační problém
  - hledáme minimální hranový řez grafu podobnosti, aby se graf $\mathcal{S}$ stal nesouvislým
    - $\min_{A \subset S} cut(A, \bar{A})$,
    - výpočetně zvladatelný problém, ale nevede na uspokojivé rozklady,



mincut, nesprávně RatioCut, Ncut, správně

  - je třeba zavést požadavek na "dostatečnou" velikost komponent
    - minimalizujeme jedno z kritérií vyváženého řezu,
    - $RatioCut(A, B) = cut(A, B)\left(\frac{1}{|A|} + \frac{1}{|B|}\right)$,
    - $Ncut(A, B) = cut(A, B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$,
    - bohužel jde o NP-těžké problémy,
- spektrální shlukování je přibližným a rychlým řešením problému vyváženého řezu.

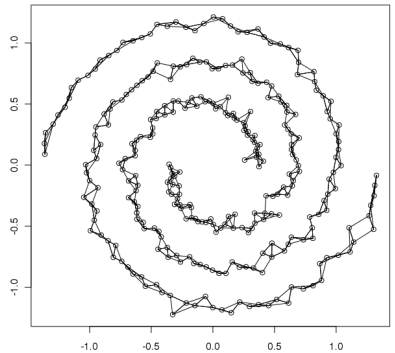
Spektrální shlukování – algoritmus

- vstupy: $\mathcal{X} = [x_{ij}]_{m \times n} = \{x_1, \ldots, x_m\} \subset \mathbb{R}^n$, k

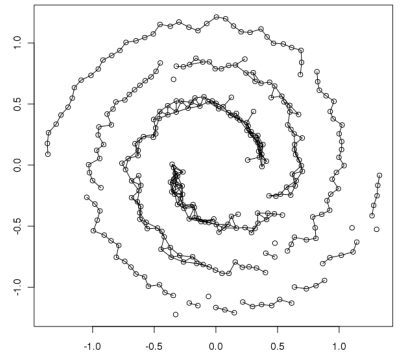
1. volba podobnostní funkce
   - lineární, RBF, polynomiální, atd.
   - neexistuje obecně platné doporučení pro přiřazení funkcí k problémům,
2. výpočet podobnostní matice $\mathcal{S} = [s_{ij}]_{m \times m}$
   - (tím vzniká i nový implicitní příznakový prostor),
3. vytvoření "rozumného" grafu podobnosti úpravami $\mathcal{S}$
   - $\mathcal{S}$ odpovídá úplnému grafu, vrcholy $\sim$ objekty, podobnosti $\sim$ hrany,
   - odstranění dlouhých (nevhodných) hran,
4. odvození Laplaceovy matice $\mathcal{L}$ z matice podobnosti $\mathcal{S}$
   - nenormalizovaná: $\mathcal{L} = \mathcal{D} - \mathcal{S}$,
   - normalizovaná: $\mathcal{L}_{rw} = \mathcal{D}^{-1}\mathcal{L} = \mathcal{I} - \mathcal{D}^{-1}\mathcal{S}$,
5. projekce do explicitního prostoru k nejmenších vlastních vektorů odvozené $\mathcal{L}$,
   - $\mathcal{V} = [v_{ij}]_{m \times k}$, vlastní vektory $\mathcal{L}$ jako sloupce,
6. shlukování pomocí k-středů v matici $\mathcal{V}$
   - řádky $\mathcal{V}$ interpretovány jako nové obrazy objektů v k-rozměrném prostoru.

Spektrální shlukování – graf podobnosti

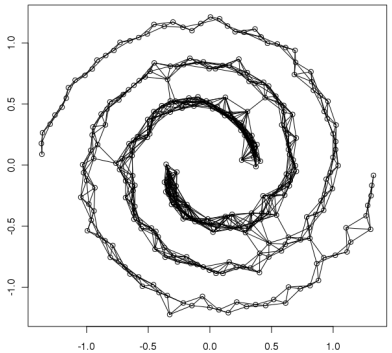
- redukce úplného grafu na neorientovaný graf vyjadřující lokalitu okolí,
- vrcholy mají rozumný stupeň ($\ll m$),
- základní přístupy
  - ϵ-okolí
    - $s_{ij} > \epsilon \rightarrow$ uzly i a j propojeny hranou, jinak $s_{ij} = 0$,
  - k-nejbližších sousedů
    - symetrický: spoj i a j pokud i je mezi k nejbližšími sousedy j nebo naopak,
    - oboustranný: spoj i a j pokud i je mezi k nejbližšími sousedy j i naopak,
  - ponechat úplný graf
    - obvykle v kombinaci s RBF nebo jiným přísně lokálním kernelem.



Euklidovská podobnost, 3 nejbližší sousedi (var A)



RBF kernel, příliš malé ϵ



RBF kernel, vhodné ϵ

Spektrální shlukování – Laplacián grafu

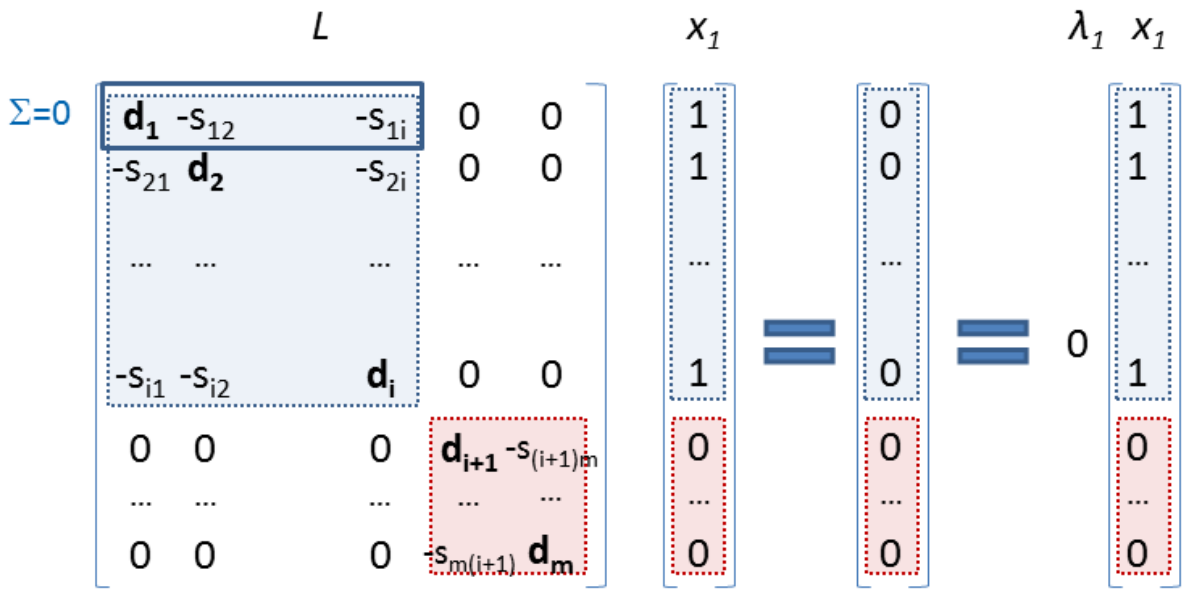
- uvažujme nenormalizovanou variantu: $\mathcal{L} = \mathcal{D} - \mathcal{S}$
- pak pro $\forall f \in \mathbb{R}^m$

$$
\begin{aligned}
f'\mathcal{L}f &= f'\mathcal{D}f - f'\mathcal{S}f = \\
&= \sum_{i=1}^{m} d_i f_i^2 - \sum_{i,j=1}^{m} f_i f_j s_{ij} = \\
&= \frac{1}{2}\Big(\sum_{i=1}^{m}(\sum_{j=1}^{m} s_{ij}) f_i^2 - 2\sum_{i,j=1}^{m} f_i f_j s_{ij} + \sum_{j=1}^{m}(\sum_{i=1}^{m} s_{ij}) f_j^2\Big) = \\
&= \frac{1}{2}\sum_{i,j=1}^{m} s_{ij}(f_i - f_j)^2
\end{aligned}
$$

- měří změnu funkce f podél grafu
  - hodnota $f'\mathcal{L}f$ je nízká pokud body blízké v grafu mají i blízké hodnoty f_i,
  - kóduje předpoklad, že blízké objekty by měly mít i podobné označení (fce f),
- podobné vlastnosti má diskrétní Laplaceův operátor,
- zajímavý případ: $f = \mathbb{1}_A$ ($f_i = 1$ if $v_i \in A$ jinak $f_i = 0$), A je komponentou grafu.

Spektrální shlukování – vlastní vektory $\mathcal{L}$

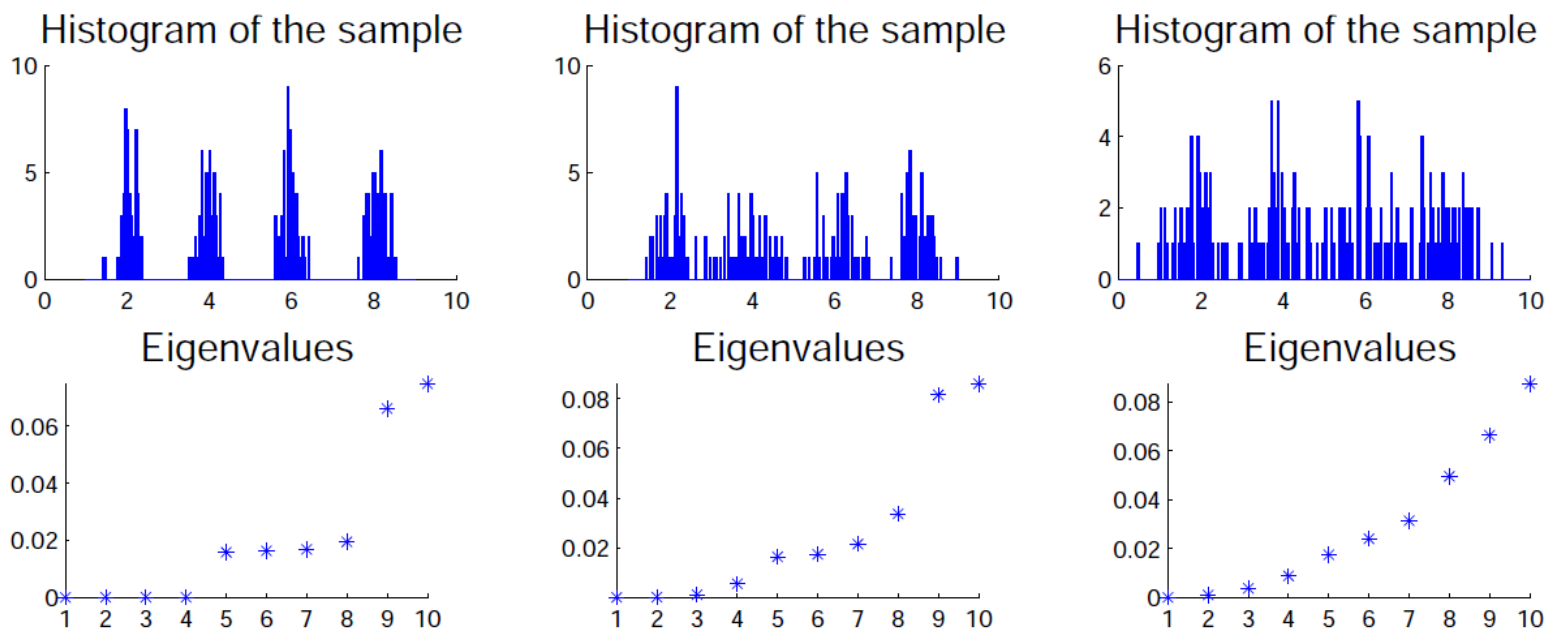
- vlastní vektory x matice $\mathcal{L}$ ($\mathcal{L}x = \lambda x$) jsou dobrými indikátory pro rozklad grafu,
- mezní (ideální) případ: graf má právě k komponent
  - k nejmenších vlastních vektorů ideálně odděluje k shluků,
  - $\lambda_1 = \cdots = \lambda_k = 0 < \lambda_{k+1} \leq \cdots \leq \lambda_m \rightarrow x_1, \ldots, x_k$,
- ostatní (obvyklé) případy: graf je souvislý, ale kandidáti k komponent existují
  - prostor k nejmenších vlastních vektorů (už s nenulovými λ) umožňuje oddělení k shluků.



Ideální případ pro $k = 2$.

Spektrální shlukování – vlastní čísla $\mathcal{L}$

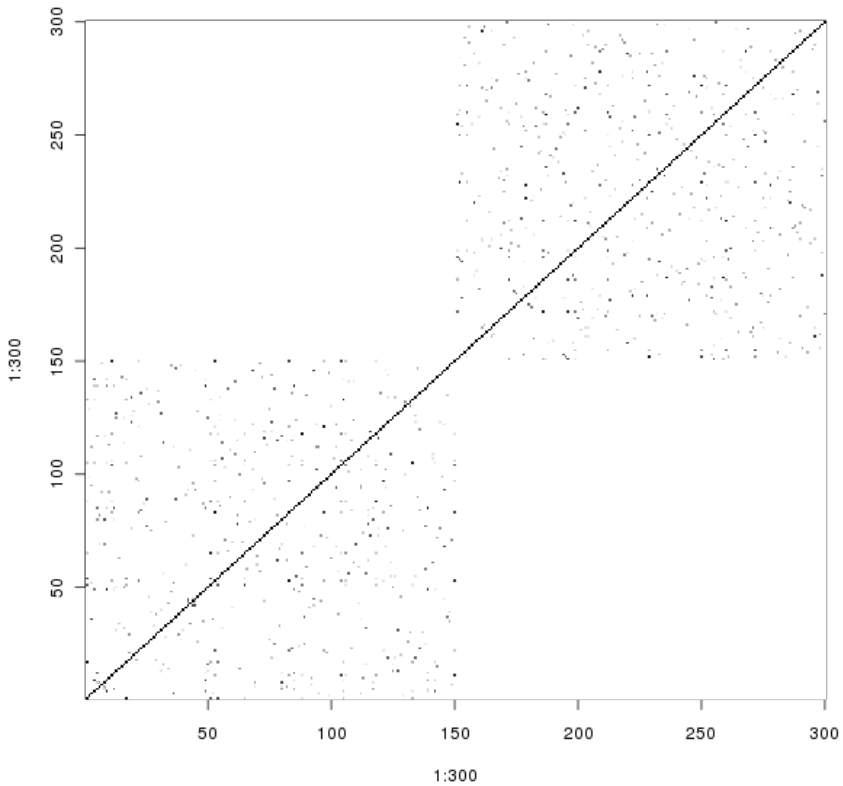
- pokud k není známo, **eigengap** statistika
  - analogie gap heuristiky u k-středů,
  - použij malé vlastní vektory do první skokové změny vlastního čísla,
  - počet shluků následně shodný se zvoleným počtem vlastních vektorů.



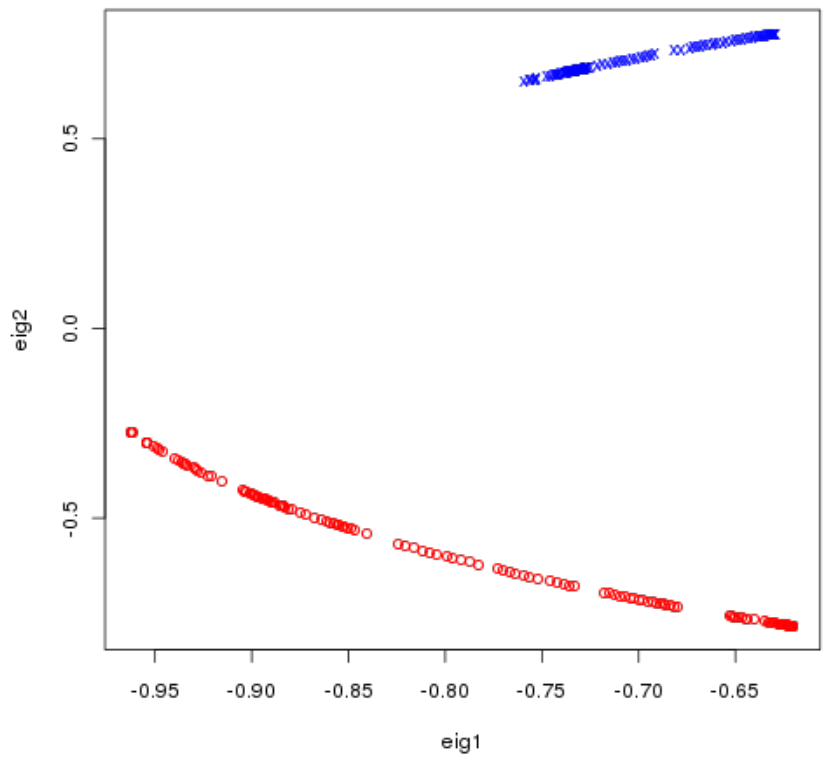
Luxburg: Clustering.

Př. spirály – vlastní vektory

- matice podobnosti dělí graf na komponenty téměř ideálně,
- druhý vlastní vektor $\mathcal{L}$ je dokonalým ukazatelem komponent.



Matice podobnosti dle RBF kernelu s vhodným σ
řazení bodů dle skutečné příslušnosti ke spirálám pro přehlednost



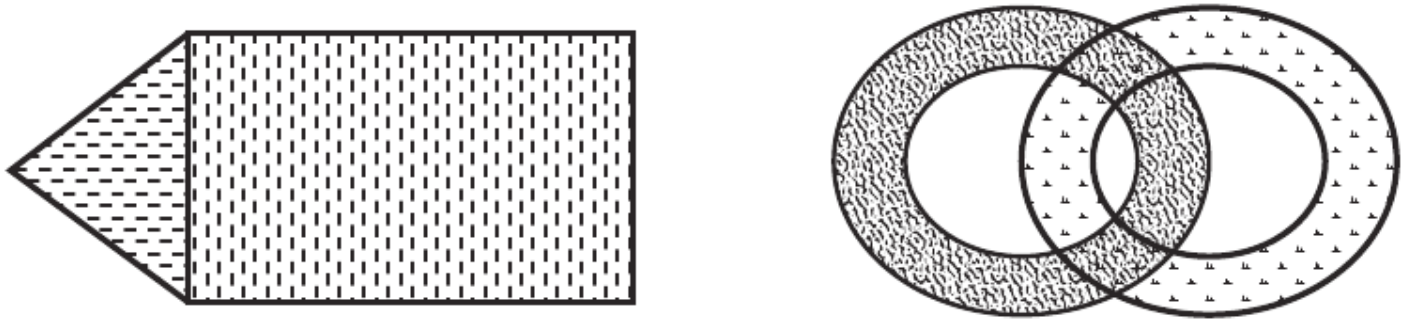
Projekce do prostoru prvních dvou vlastních vektorů $\mathcal{S}$
barvy dle skutečné příslušnosti ke spirálám, řešení pomocí k-středů triviální

Spektrální shlukování – shrnutí

- výhody
  - nepředjímá konkrétní tvar shluků,
  - jednoduchá implementace – využití existujících algoritmů,
  - nehrozí uváznutí v lokálním optimu,
  - modulární přístup použitelný pro širokou škálu problémů
    - modifikace kernelu nebo určení grafu podobnosti adaptují na úlohu,
  - eigengap heuristika pro nalezení optimálního počtu shluků,
  - prokazatelné úspěchy v řadě aplikací,
- nevýhody
  - neexistuje jasný návod pro ideální volbu parametrů
    - kernely (mj. σ u RBF), grafy podobnosti (ϵ nebo k),
  - výpočetně náročné pro velké grafy, které nejsou řídké,
    - použití teprve pokud jednodušší algoritmy selžou,
  - nejasný smysl a funkce pro neregulární grafy,
- demo
  - http://www.ml.uni-saarland.de/GraphDemo/GraphDemo.html.

Konceptuální shlukování – motivace

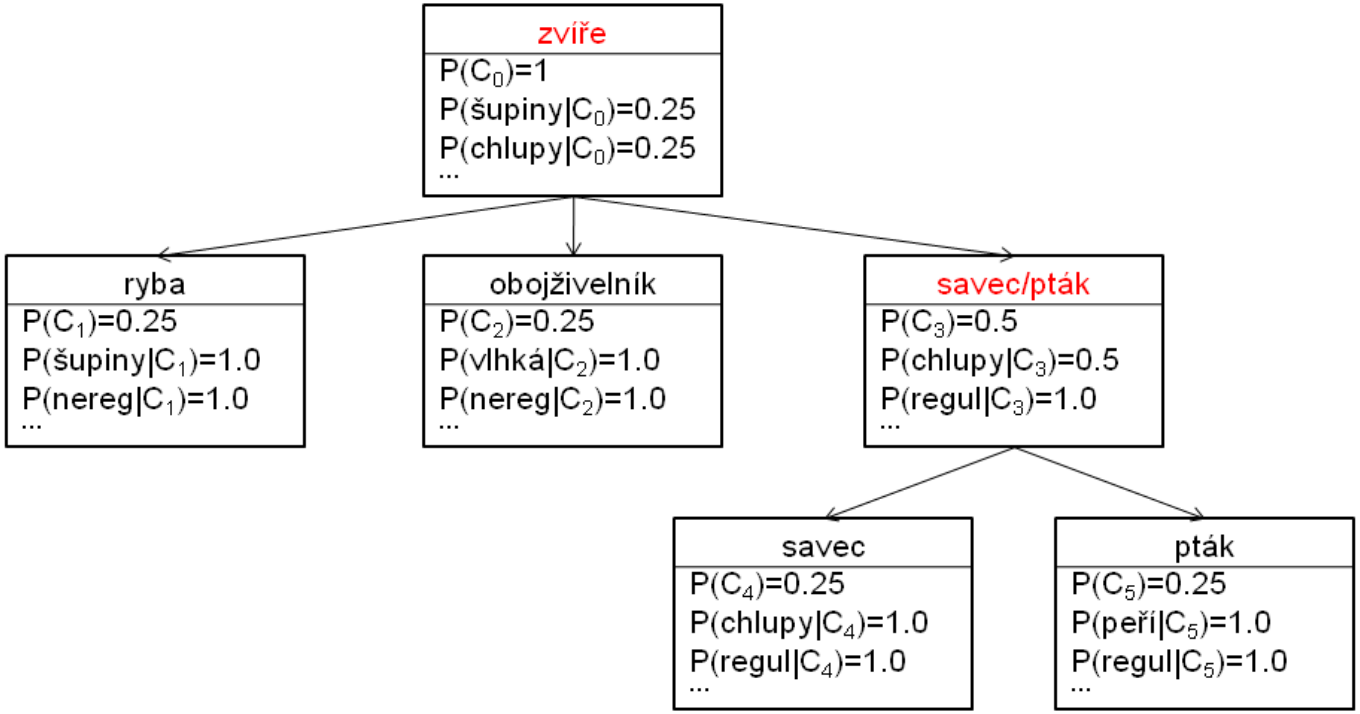
- shluk tvoří objekty, které sdílí nějakou **vlastnost**
  - zobecňuje dřívější definici založenou na podobnosti
  - (tam byla společná vlastnost např. blízkost ke stejnému středu),
- kromě vlastního rozkladu generuje i **popis** shluků
  - koncept = pojem (= definice v nějakém **jazyce**),
  - (je to velké, žluté, dělá to rámus, rychlé, vejde se do toho 5 lidí . . . je to auto),
- na pomezí shlukování a rozpoznávání
  - pokud je vlastnost definující shluk příliš komplikovaná.



Tan et al.: Introduction to Data Mining

Konceptuální shlukování: COBWEB (Fisher, 1987)

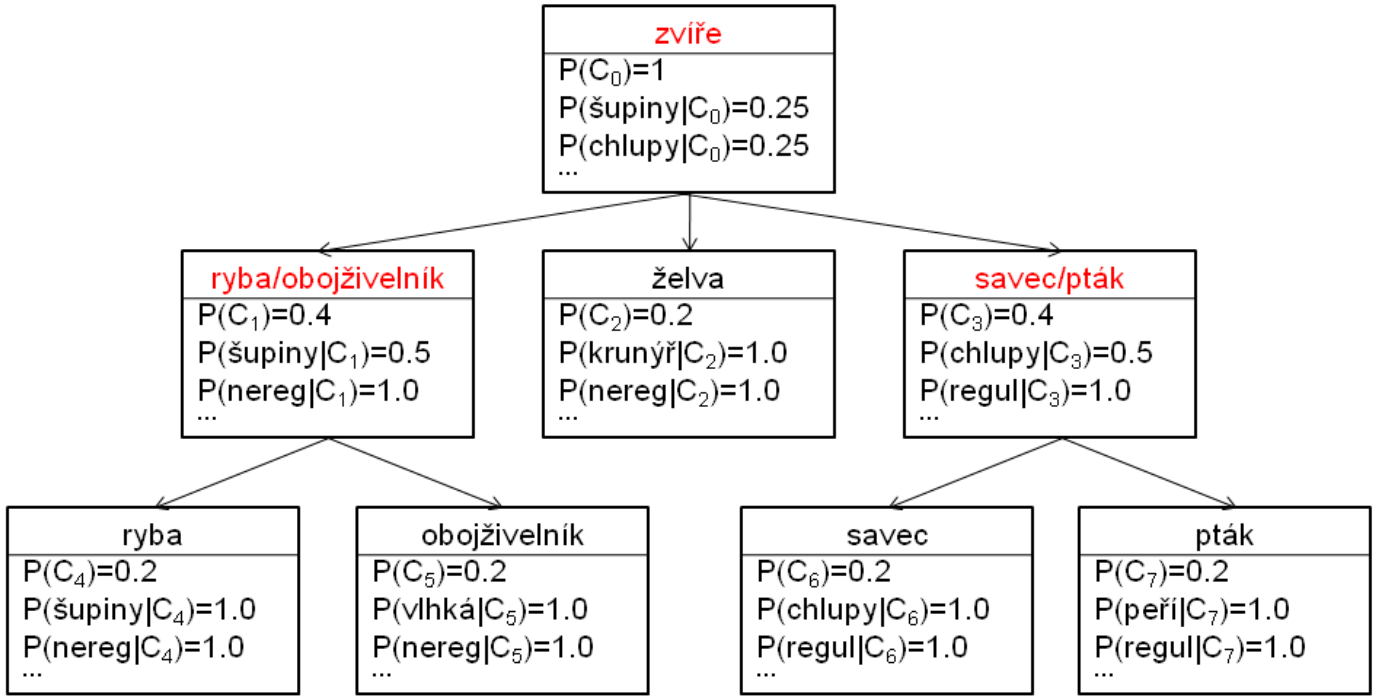
Jméno	Povrch	Komory	Teplota	Oplození
ryba	šupiny	dvě	neregulovaná	vnější
obojživelník	vlhká_kůže	tři	neregulovaná	vnější
savec	chlupy	čtyři	regulovaná	vnitřní
pták	peří	čtyři	regulovaná	vnitřní
želva	krunýř	čtyři_nedok	neregulovaná	vnitřní



Fisher: Konceptual Acquisition Via Incremental Conceptual Clustering

Konceptuální shlukování: COBWEB (Fisher, 1987)

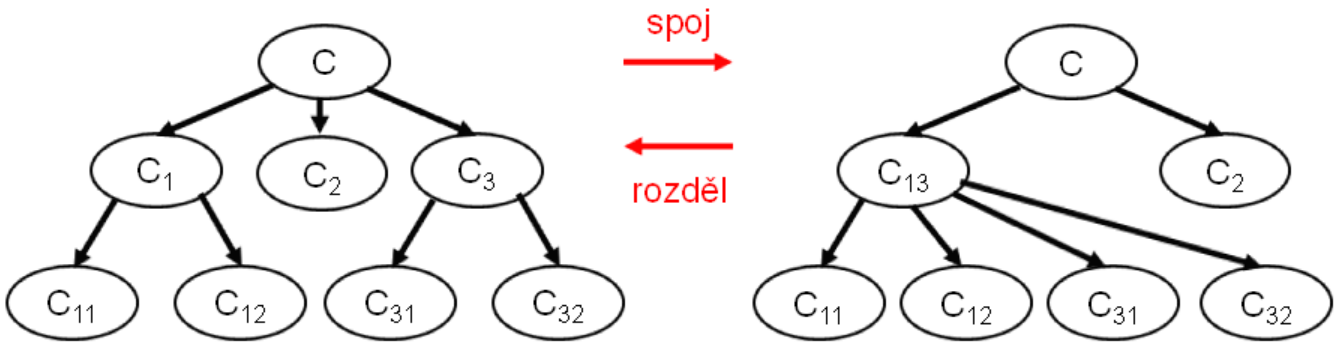
Jméno	Povrch	Komory	Teplota	Oplození
ryba	šupiny	dvě	neregulovaná	vnější
obojživelník	vlhká_kůže	tři	neregulovaná	vnější
savec	chlupy	čtyři	regulovaná	vnitřní
pták	peří	čtyři	regulovaná	vnitřní
želva	krunýř	čtyři_nedok	neregulovaná	vnitřní



Fisher: Konceptual Acquisition Via Incremental Conceptual Clustering

COBWEB algoritmus

- příklad po příkladu tvoří (klasifikační) strom,
  - vnitřní uzel = (pravděpodobnostní) koncept, list = příklad,
- pro každý jednotlivý příklad volíme jednu z těchto operací
  - vytvoř novou třídu – najdi pozici v hierarchii pro novou podtřídu,
  - zařaď příklad do existující třídy – objekt je podobný objektům v jedné ze tříd,
- abychom minimalizovali vliv pořadí příkladů, zvážíme i operace
  - spojení tříd – dvě třídy nahrazeny jedinou,
  - rozdělení třídy – jedna třída se rozpadá na podtřídy, resp. jednotlivé objekty,
- obousměrné gradientní prohledávání řízené funkcí kategoriální užitečnosti.



COBWEB: hodnotící kritérium

- Kdy je shlukování kvalitní?
  - hodnoty příznaků jsou **předvídatelné** uvnitř tříd
    - "Je to taxi." $\Rightarrow$ "Je to žluté.",
    - sledujeme psti $Pr(f_i = v_{ij}|C_c)$ (zde je vysoká $Pr(f_{barva} = zluta|C_{taxi})$)
      - $F = \{f_1, \ldots, f_n\} \ldots$ množina příznaků,
      - $V_i = \{v_{i1}, \ldots, v_{il}\} \ldots$ množina hodnot i-tého příznaku,
      - $\Omega = \{C_1, \ldots, C_k\} \ldots$ rozklad množiny $\mathcal{X}$,
    - odpovídá homogenitě uváděné dříve,
  - hodnoty příznaků jsou **určující** pro třídy
    - "Je to chlupaté." $\Rightarrow$ "Je to savec.",
    - sledujeme psti $Pr(C_c|f_i = v_{ij})$ (zde je vysoká $Pr(C_{savec}|f_{povrch} = chlupaty)$),
    - odpovídá separabilitě uváděné dříve,
  - výše uvedené platí zejména pro **časté** hodnoty příznaků
    - "Používá to k orientaci ultrazvuk." $\Rightarrow$ "Je to savec.",
    - málo užitečné, protože $Pr(f_{orientace} = ultrazvuk) \rightarrow 0$,
  - rozklad je stručný
    - čím méně kategorií, tím lépe.

COBWEB: hodnotící kritérium

- COBWEB jako míru kvality používá tzv. **kategoriální užitečnost**
  - shrnuje vlastnosti uvedené na předchozím slajdu,
  - jde o variantu vzájemné informace,
- kvalita rozkladu – častost příznaků $\times$ určitelnost tříd $\times$ předvídatelnost příznaků

$$\sum_{c=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{|V_i|} Pr(f_i = v_{ij})Pr(C_c|f_i = v_{ij})Pr(f_i = v_{ij}|C_c) =$$

$$= \sum_{c=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{|V_i|} Pr(C_c)Pr(f_i = v_{ij}|C_c)Pr(f_i = v_{ij}|C_c) = \sum_{c=1}^{k} Pr(C_c)\sum_{i=1}^{n}\sum_{j=1}^{|V_i|} Pr(f_i = v_{ij}|C_c)^2$$

- kategoriální užitečnost po zahrnutí
  - stručnosti rozkladu – normalizace počtem shluků,
  - referenční informace v datech bez jakékoli informace o shlucích.

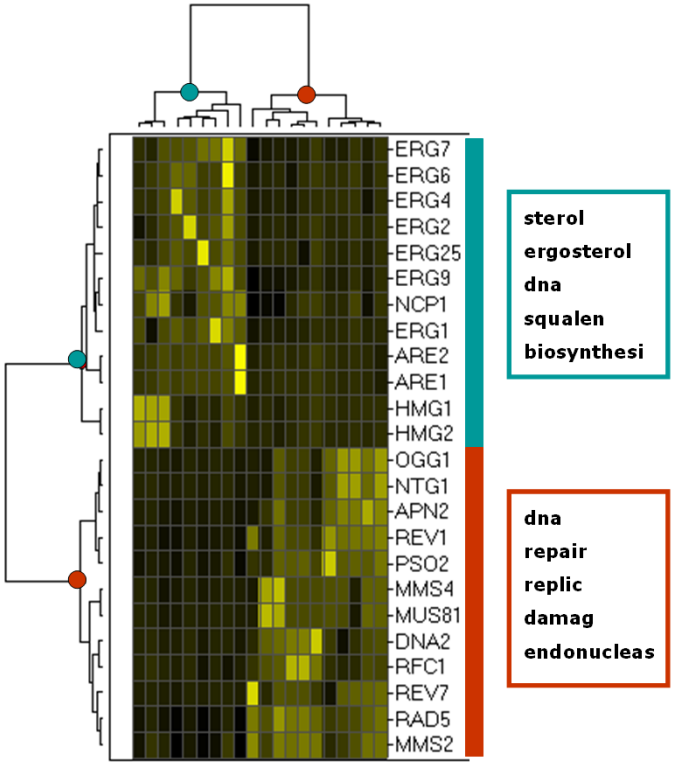
$$UC(\Omega, \mathcal{X}) = \frac{1}{k}\sum_{c=1}^{k} Pr(C_c)\Big(\sum_{i=1}^{n}\sum_{j=1}^{|V_i|} Pr(f_i = v_{ij}|C_c)^2 - \sum_{i=1}^{n}\sum_{j=1}^{|V_i|} Pr(f_i = v_{ij})^2\Big)$$

COBWEB: shrnutí

- vlastnosti algoritmu
  - symbolický: nominální příznaky,
  - hierarchický: tvoří taxonomii,
  - inkrementální: výstup upravuje postupně, instanci po instanci,
- s jakým **jazykem** pracuje?
  - pravděpodobnostní popis konceptů je jazykem jen ve volném smyslu,
  - umožňuje zápis libovolného konceptu,
    - příklad pro tři binární příznaky: $Pr(x|C_j) = [0.6, 0.5, 0.7]$,
  - lze jej rozšířit o omezení – pak jde o silnější jazyk
    - definujeme činitel α, ten určuje mininimální odchylku od 0.5,
    - omezení může platit pro největší odchylku nebo pro všechny odchylky,
    - $\alpha = 0.3$ alespoň pro jeden příznak
      - vektor $[0.6, 0.5, 0.7]$ neplatným konceptem, $[0.6, 0.5, 0.9]$ platným,
    - limitní případ $\alpha = 0.5$ pro všechny příznaky
      - platné pouze logické konjunktivní koncepty.

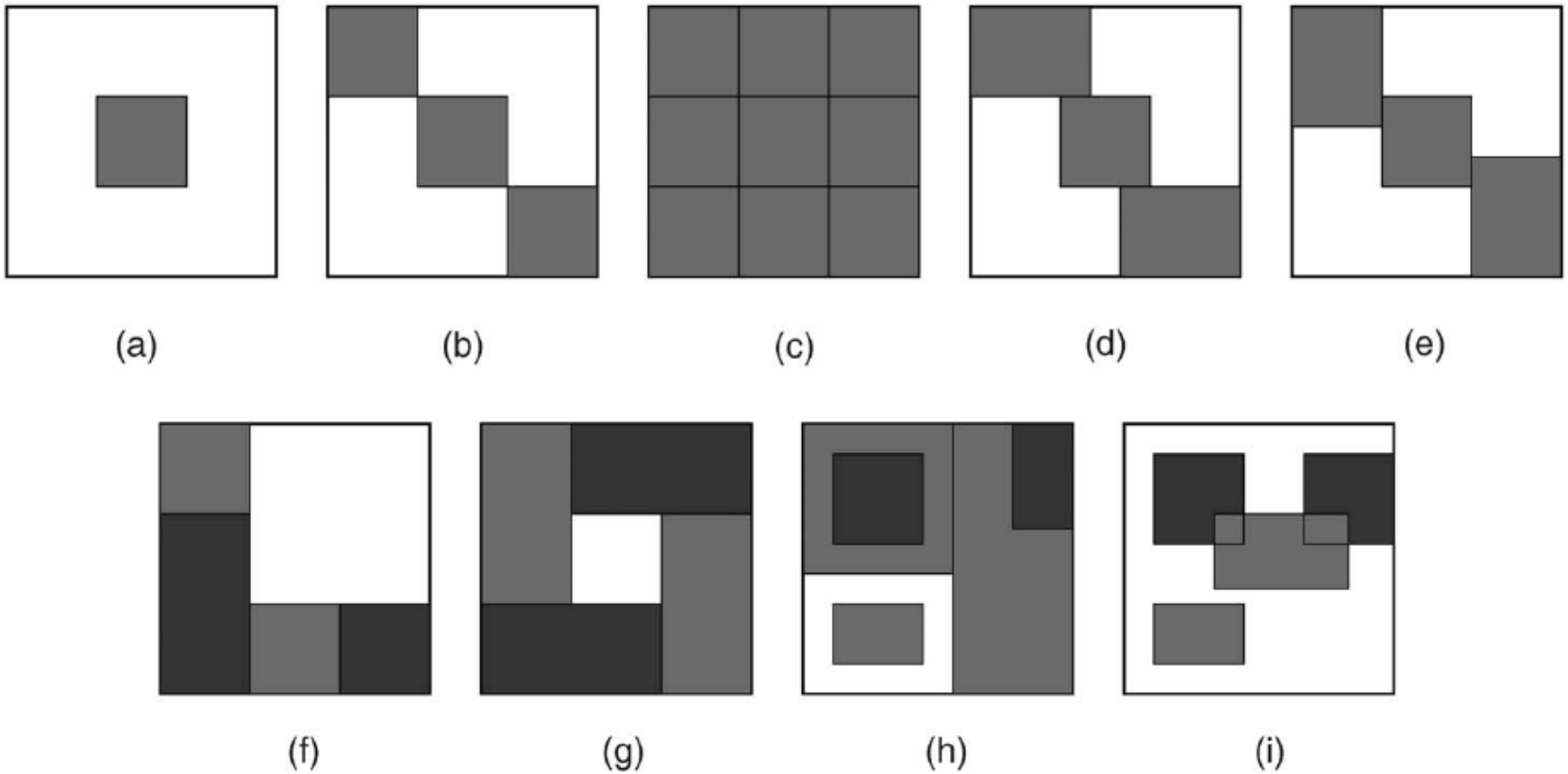
Dvojshlukování (biclustering)

- (hierarchické) shlukování instancí i příznaků,
- častá aplikace v analýze genové exprese,
  - instance: tkáně, příznaky: aktivity genů
- na obrázku shlukování odděleně v každém z rozměrů
  - napomáhá přehlednosti zobrazení,
  - jde o globální model
    - ∗ všechny tkáně použity při shlukování genů,
    - ∗ všechny geny použity pro shlukování tkání,
- dvojshlukování shlukuje **současně** v obou rozměrech
  - jde o lokální model
    - ∗ pouze důležité tkáně určují konkrétní shluk genů,
    - ∗ pouze důležité geny určují konkrétní shluk tkání,
  - více možných funkcí genů za různých podmínek,
  - není výlučné ani úplné (překryvy, vynechání).



Dvojshlukování – struktura shluků

(a) jediný dvojshluk, (b) výlučné dvojshluky, (c) šachovnice, (d) výlučné řádky, (e) výlučné sloupce, (f) hierarchické nepřekrývající se, (g) nepřekrývající se nevýlučné, (h) hierarchické překrývající se, (i) volně umístěné překrývající se.



Madeira, Oliveira: Biclustering Algorithms for Biological Data Analysis.

Dvojshlukování – příklad algoritmu

- **block clustering** (také direct clustering, Hartigan)

  1. seřaď řádky (sloupce) dle řádkového (sloupcového) průměru,
  2. najdi nejlepší řádkové či sloupcové dělení,
     (a) kritériem je pokles rozptylu uvnitř bloku, resp. celkového kritéria

$$\sum_{c=1}^{k} \sum_{i \in I, j \in J} (x_{ij} - x_{IJ})^2$$

     (I, J) je dvojshluk, $I \subseteq \{1 \dots m\}$, $I \subseteq \{1 \dots n\}$, x_{IJ} je střední hodnota v dvojshluku,

     (b) dělení pouze za daného uspořádání – lineární počet testů s m i n,
  3. opakuj předchozí krok v obou vzniklých podblocích dokud není dosaženo zvolené k,

  – hierarchický přístup rozděl a panuj,
  – triviální, rychlý algoritmus se sklonem k hrubě suboptimálním řešením,
  – hledá dvojshluky typu (f), tj. hierarchické nepřekrývající se,
  – upřednostňuje konstantní dvojshluky – ideálem je identická hodnota v celém dvojshluku.

Block clustering – US prezidentské volby

- procento hlasů odevzdaných pro republikánského kandidáta na prezidenta,
- jižanské státy v letech 1900-1968.

State	Year																	
	12	36	32	40	44	48	16	04	68	08	24	00	20	28	56	60	52	64
SC	1	1	2	4	4	4	2	5	39	6	2	7	4	9	25	49	49	59
MI	2	3	4	4	6	3	5	5	14	7	8	10	14	18	24	25	40	87
GA	4	13	8	15	18	18	7	18	30	31	18	29	29	45	33	37	30	54
LA	5	11	7	14	19	17	7	10	23	12	20	21	31	24	53	29	47	57
AA	8	13	14	14	18	19	22	21	14	24	27	35	31	48	39	42	35	70
TS	9	12	11	19	17	25	17	22	40	22	20	31	24	52	55	49	53	37
FA	8	24	25	26	30	34	18	21	41	22	28	19	31	57	57	52	55	48
AS	20	18	13	21	30	21	28	40	31	37	29	35	35	39	46	43	44	44
VA	17	29	30	32	37	41	32	37	43	38	33	44	38	54	55	52	56	46
NC	12	27	29	26	33	33	42	40	40	46	40	45	43	55	49	48	46	44
TE	24	31	32	33	39	37	43	43	38	46	44	45	51	54	49	53	50	44
KY	25	40	40	42	45	41	47	47	44	48	49	49	49	59	54	54	50	36
MD	24	37	36	41	48	49	45	49	42	49	45	52	55	57	60	46	55	35
MO	30	38	35	48	48	42	47	50	45	49	50	46	55	56	50	50	51	36
WV	21	39	44	43	45	42	49	55	40	53	49	54	55	58	54	47	48	32
DE	33	43	51	45	45	50	50	54	45	52	58	54	56	65	55	49	52	39

Hartigan: Direct Clustering of a Data Matrix.

Dvojshlukování – příklad algoritmu

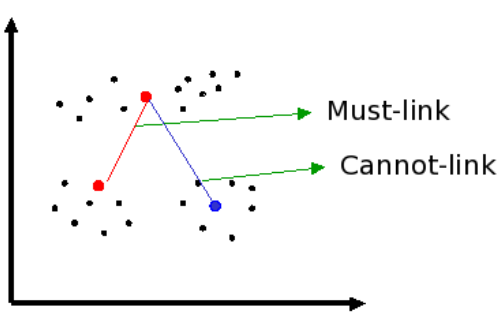
- **Cheng a Church**
  - první algoritmus použitý pro dvojshlukování dat z genových čipů,
  - používá upravenou definici rozptylu (obecně reziduí)
    $\sum_{c=1}^{k} \frac{1}{|I||J|} \sum_{i \in I, j \in J} (x_{ij} - x_{Ij} - x_{iJ} + x_{IJ})^2$
    * superponujeme pozadí (x_{IJ}), efekt genu (x_{iJ}) a efekt biologických podmínek (x_{Ij}),
  - problém: některé triviální dvojshluky mají nulové reziduum ($1 \times m$ nebo $1 \times n$)
    * zavádí prahové reziduum δ a hledá největší dvojshluky s reziduem$< \delta$,
  - hladový algoritmus (hledá dvojshluky typu (i))
    1. začni s celou maticí,
    2. odstraň řádek či sloupec s největším poklesem rezidua,
    3. opakuj krok 2, skonči pokud reziduum nejde snížit nebo je $< \delta$,
    4. hodnoty v nalezeném dvojshluku znáhodni, opakuj (1-3) pro k dvojshluků.
- obrázek: aditivní (vlevo) a multiplikativní (vpravo) dvojshluk.

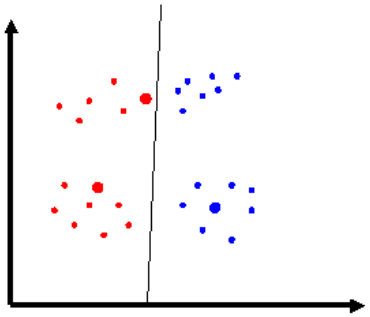
1.0	2.0	5.0	0.0
2.0	3.0	6.0	1.0
4.0	5.0	8.0	3.0
5.0	6.0	9.0	4.0

1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

Shlukování s poloučitelem (semisupervised)

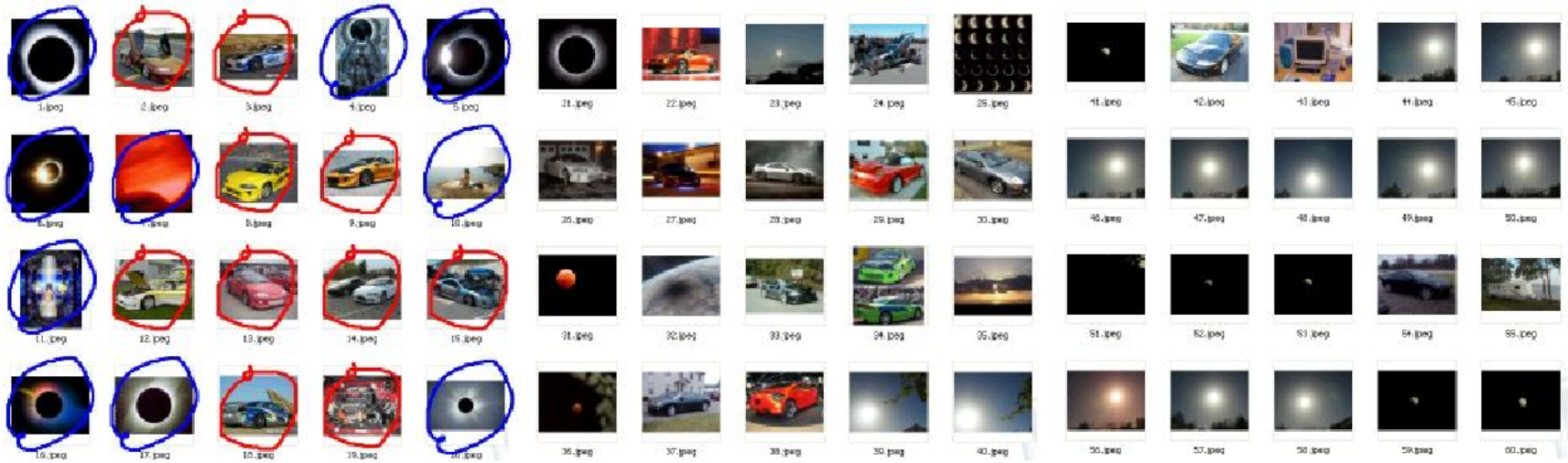
- na vstupu kromě neoznačených příkladů také apriorní znalost
  - málo označených příkladů (zárodky shluků),
  - omezení (obvykle typu **must-link** nebo **cannot-link**),
- úpravy stávajících algoritmů
  - EM shlukování pro klasifikované i neklasifikované příklady
    1. zkonstruuj model pro pravděpodobnostní klasifikaci z počátečních klasifikovaných dat,
    2. standardní EM iterace, počátečně klasifikované příklady jsou "zmrazeny" (tj. nejsou překlasifikovávány),
  - úprava kvalitativního hodnocení rozkladu
    - kromě homogenity i odměna za dodržení must-link a penalta za porušení cannot-link,





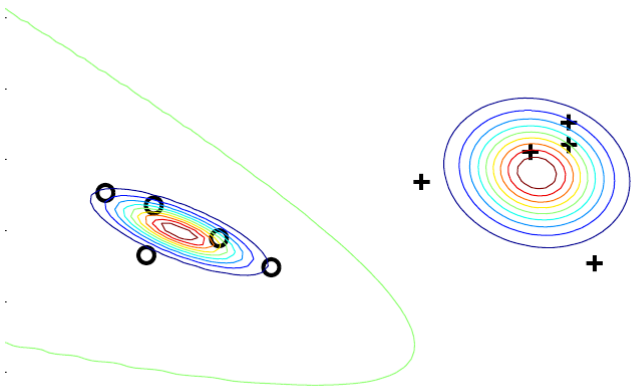
Shlukování s poloučitelem (semisupervised)

- jak se naučit rozpoznávat fotografie zatmění za rozumnou dobu?

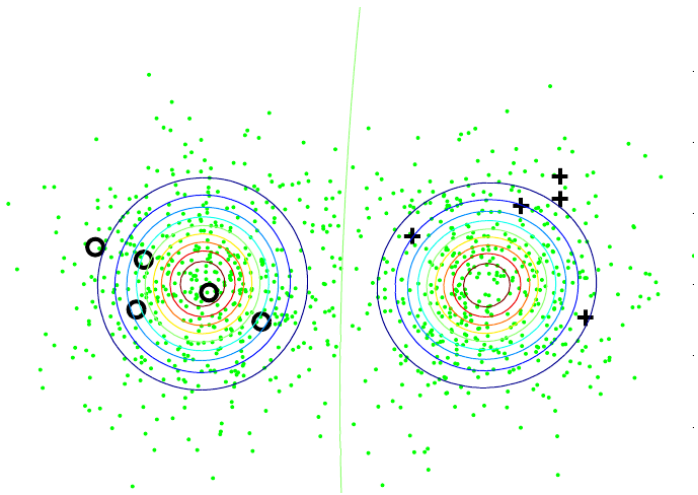


Shlukování s poloučitelem (semisupervised)

- př.: úspěšné použití pro klasifikaci dokumentů do 20 diskusních skupin
  - k dipozici pouze 2 klasifikované trénovací příklady pro každou skupinu,
  - dále 10000 neoznačených příkladů,
  - naivní Bayesův model ze 40 dokumentů (bez shlukování): 27% klasifikační přesnost,
  - stejný model z 10040 dokumentů (po shlukování): 43% klasifikační přesnost,
  - zásadní pro úspěch: shluky v datech musí odpovídat třídám.
- jak lze zlepšení klasifikační přesnosti vysvětlit?



Rozhodovací hranice pouze pro značené instance



Rozhodovací hranice s neoznačenými instancemi

Zhu: Semi-Supervised Learning Tutorial

Pokročilé metody shlukování – shrnutí

- Shlukování je široký pojem
  - nejenom rozklad množiny příkladů v $\mathbb{R}^n$ na disjunktní shluky,
  - obecně identifikace jakéhokoli častého spoluvýskytu událostí v datech,
- jde o subjektivní postup
  - neexistuje "základní pravda" jako u klasifikace,
  - o výsledku rozhodují použité implicitní i explicitní znalosti,
- úzce se váže na učení
  - konceptuální shlukování
    - znalostní shlukování s popisem výsledných shluků/pojmů,
  - učení s poloučitelem
    - shluky či klasifikace na základě anotovaných i neanotovaných příkladů,
- existuje v řadě modifikací
  - dvojshlukování
    - příklon od globální k lokální podobnosti.

Doporučené doplňky – zdroje přednášky

:: Četba

- von Luxburg: **Lectures on Clustering.**
  - PASCAL Bootcamp in Machine Learning, Vilanova (Barcelona), 2007,
  - http://videolectures.net/bootcamp07_luxburg_clu/,
- Fisher: **Knowledge Acquisition Via Incremental Conceptual Clustering.**
  - Machine Learning 2, 139-172, 1987.
  - http://www.springerlink.com/index/X8552PPN35245112.pdf,
- Madeira, Oliveira: **Biclustering Algorithms for Biological Data Analysis.**
  - IEEE/ACM Trans Comput Biol Bioinform. Jan-Mar 1(1):24-45, 2004.
  - http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.3943,
- Zhu: **Semi-Supervised Learning Tutorial.**
  - http://pages.cs.wisc.edu/~jerryzhu/pub/sslicml07.pdf.
- Nigam, McCallum, Mitchell: **Semi-Supervised Text Classification Using EM.**
  - In Chapelle, Zien, Scholkopf (eds.) Semi-Supervised Learning. MIT Press, 2006.