

TABULKOVÁ DATA

- Abstraktní data – nemá smysl je nějak obohatovat
- Struktura:
 - o položka/item – řádek tabulky
 - o atribut – sloupec tabulky
 - o buňka – průsečík řádku se sloupcem (hodnota atributu v rámci položky)
- **Počet atributů** je brán jako **počet dimenzí**
 - o Položka (řádek tabulky) je bod v n-dimenzionálním prostoru
- Příklady:
 - o Výsledky dotazníků – položka je jeden vyplněný dotazník, atributy/dimenze jsou otázky v dotazníku, hodnoty jsou odpovědi na otázky
 - o Data, co popisují vlastnosti objektů, nebo služeb – vlastnosti auta, vlastnosti life insurance plans
 - o Data z několika senzorů – měrné hodnoty hvězd v naší galaxie (každou vlastnost měříme jiným senzorem)

Enkódování dat na vizuální kanály

- Na **pozici** – je to nejvíce přesné
 - o Různé layouty os jsou možné (ortogonální osy jsou časté – kartézský souřadnicový systém, pararelní osy, ...)
- Na **tvar**
 - o Většinou máme velké množství atributů, takže potřebujeme enkódovat data na geometrii, která bude měnit tvar se změnou dat – glyfy

Typické tasky

- Identifikační task – target je 1 atribut
 - o Vizualizace vyjadřuje hodnotu nebo range, rozložení hodnot pro individuální atributy
 - o Jaká je hodnota atributu X pro danou položku?
 - o Jaké je rozložení položek pro daný atribut
- Identifikační task – target jsou položky
 - o Jaké položky mají hodnotu X pro daný atribut?
 - o Jaké položky mají hodnotu daného atributu v rangi [X, Y]
- Identifikační task – target je mnoho atributů
 - o Vizualizace ukazuje relace, vztahy mezi atributy
 - o Je tam nějaká korelace či dependency mezi atributy?
 - o Formují položky clustery na základě hodnot několika atributů?
 - o Task je náročnější s přibývajícím množstvím atributů

Motivace pro vizualizaci tabulkových dat

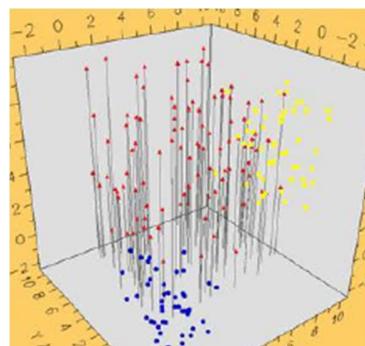
Pro 2D scatter plot – 2 atributy

- Není žádný problém

Po přidání dalšího atributu lze rozšířit scatterplot na 3D

- Problém ale je, že v tom 3D scatterplotu nevidíme moc dobře, kde ty body leží
- Pomáhá interakce – rotace, ...

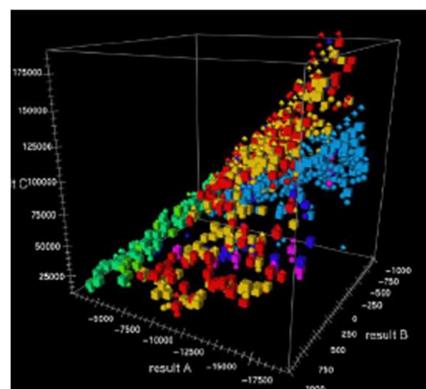
- Můžeme přidat dodatečné informace, např. vertikální čáry od bodů (od červených bodů)
 - o Může ale lehce vzniknout visual clutter



- Do jisté míry funguje, ale jsou tam problémy

Při dalším zvýšení dimenze

- Lze využít enkódování na barvu a na velikost nebo tvar
- Šílený visual cluster



- Pro tabulková data je velmi důležitá **interakce**, zejména:
 - o Manipulace – selekce a zvýrazňování
 - o Redukce – Filtrování, agregace
 - o Organizace pohledů – Juxtaposition, Brushing and linking

METODY VHODNÉ PRO KVANTITATIVNÍ SPOJITÉ ATRIBUTY:

FACETING

- Menší množství atributů
- Vizualizujeme každou kombinaci atributů
- Uspořádáme vizualizaci na obrazovce
- Faceting redukuje prostor na obrazovce s každou kombinací atributů, takže není použitelný pro vyšší počet atributů

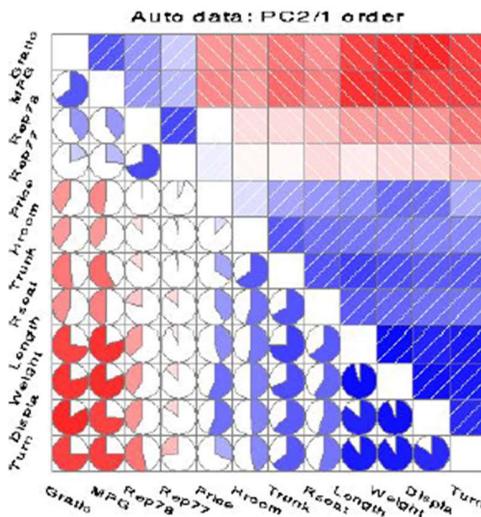
Např. **Scatterplot matrix**

- Zobrazí scatterplot pro všechny páry atributů
- Clustery a korelace mohou být lehce vidět pro páry atributů
- Ale clustery a korelace mezi více než dvěma atributy je těžké vidět

Interakce:

- Brushing (selection) by dovolil linkage mezi scatterploty – to by nám pomohlo vidět clustery, trendy, korelace mezi více než dvěma atributy
- Často se používá možnost, že uživatel vybere nějakou obdélníkovou oblast v jednom scatterplotu a položky jsou pak v této oblasti zvýrazněné ve všech scatterplotech (linkage)
- Uživatel může použít více brushes, aby zvýraznil odlišné položky s odlišnými barvami

Pokud je těch atributů hodně a scatterploty by byly příliš malé, aby z nich šlo rozumě vykoukat informaci, tak lze použít Corgram. To zobrazuje míru korelace mezi jednotlivými atributy.

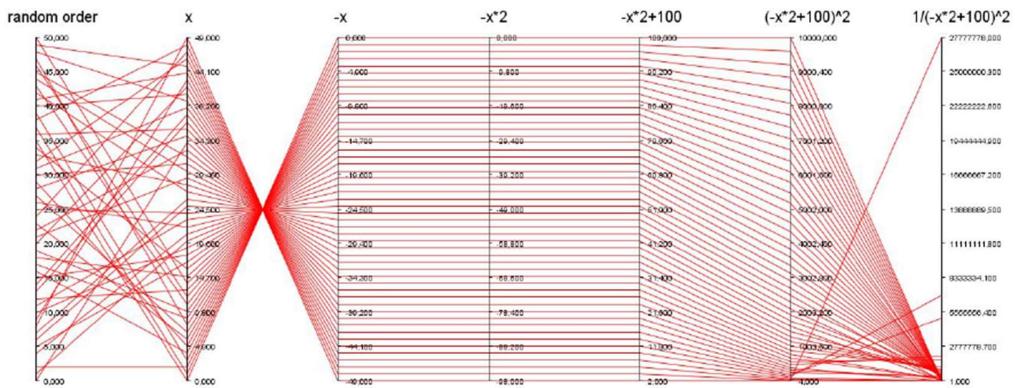


PARARELNÍ SOUŘADNICE

- Osy jsou paralelně vůči sobě (bázové vektory jsou lineárně závislé)
- Myšlenka: místo toho, abychom vykreslovali body, tak budeme vykreslovat lomennou čáru
- Vyneseeme body na pararelní osy x, y, z a pak tyto body propojíme čarou
- Lomenná čára nám dává reprezentaci bodu v n-rozměrném prostoru

Úlohy, které řeší:

- Clustery, trendy a anomálie mezi více než dvěma atributy – lze vidět
- Korelace mezi dvěma osy, ale jen těch, co jsou vedle sebe (není to ale úplně intuitivní, musíme se na to trochu nacvičit, jestli je to negativní či pozitivní korelace, atd.)
- Můžeme osy přehazovat, abychom viděli korelací mezi různými páry os
- Nemůžeme vidět korelací mezi všemi páry atributů najednou jako ve scatterplot matrixu



- + **Random order vs. x:** no correlation
- + **x vs. -x:** negative linear correlation
- + **-x vs. -x^2:** positive linear correlation (note that the multiplication has no effect)
- + **-x^2 vs. -x^2+100:** positive linear correlation (addition has also no effect)
- + **-x^2+100 vs. (-x^2+100)^2:** positive exponential correlation
- + **(-x^2+100)^2 vs. 1/(-x^2+100)^2:** negative exponential correlation

- Percepcí korelací můžeme zlepšit mapováním na barvu

Interakce:

- **Brushing** na zvýrazňování položek, co nás zajímají
 - o Brushing typicky specifikováváme intervaly: na **jedné ose** nebo na **několika osách**
 - o Položky, které protínají všechny intervaly brush jsou zvýrazněny
 - o Položky, které nejsou zvýrazněny vizuálně potlačíme
 - o Více brushes mohou být specifikovány a každý může mít odlišnou barvu
 - o Brushes jsou „dotazy“, kterými filtroujeme ta data, která chceme

Screen space problem

Velké množství atributů

- Nejsme schopní všechny paralelní osy zobrazit
- Scrollování možná pomůže, ale bude velmi těžké vypozorovat clustery mezi velkým množstvím atributů, protože na to, abychom viděli cluster potřebujeme vidět všechny atributy

Velké množství položek

- Nerozlišíme individuální položky (čáry od sebe)
- Pomáhá průhlednost, ale pro velké množství atributů to stejně nepomůže

Pro velmi velké množství dat máme následující techniky:

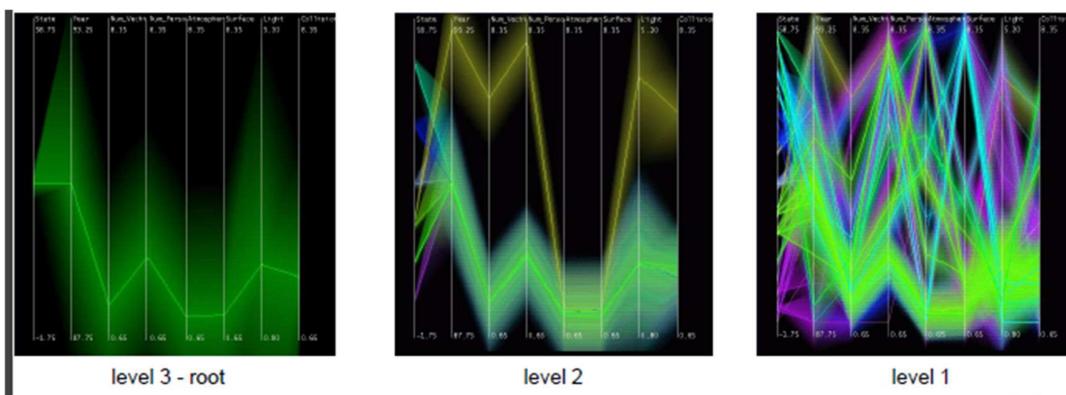
HIERARCHICAL PARALLEL COORDINATES FOR EXPLORATION OF LARGE DATASETS

- Hierarchické clustrování pro n-dimenzionální body, tím se získá hierarchie shluků
- To nám umožní prozkoumávat data na různé úrovni
- Vizualizujeme clustery na specifické úrovni clustrovací hierarchie namísto bodů
 - o Místo čar máme jednu společnou čáru odpovídající průměru hodnot bodů
 - Kolem čáry je poloprůhlednost, abychom indikovali range dat v clusteru

- Různé úrovně rozbalujeme, když se nám zdá, že je ten shluk nečím zajímavý
- Informace o outlierech nemáme

OUTLIER-PRESERVING FOCUS+CONTEXT VISUALIZATION IN PARALLEL COORDINATES

- Snaží se zachovat informaci o outlierech
- Pro každou dvojici sousedních os si vytvoříme scatterplot a ten rozdělíme do binů
- Pak počítáme kolik bodů spadne do té mřížky
- Když do mřížky spadlo hodně bodů, tak je to trend, pokud ne, tak je outlier
- Rozdělíme body v n-dimenzionálním prostoru na dvě skupiny: **outlier** a **trend**
 - Trendy reprezentujeme jako paralelogramy, které jsou mezi sebou propojeny
 - Outliery jsou jednotlivé lomenné čáry

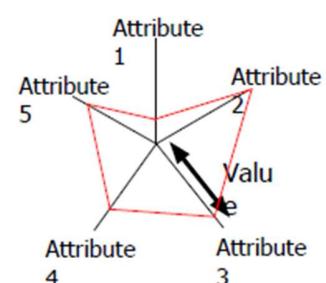


GLYFY

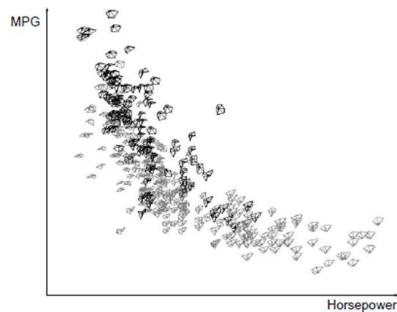
- Mohou být použity, když počet položek není velký
- Je vytvořen pro každou položku
- Mapují datové atributy na grafický element
- Mapování dat mění tvar glyfu + velikost, oblast, ... glyfu
- Podobnosti ve featurech nám mohou dát nějaké dodatečné informace o vztazích
- Mapování musí být správně uděláno:
 - Pro některé atributy je lepší, když je ho méně – kriminalita
 - Pro některé je lepší, když je ho více – mzda
 - Měli bychom rozdělit atributy na **dvě skupiny** – „více je lepší, méně je lepší“, nebo invertovat mapování pro jednu skupinu (typicky pro skupinu „méně je lepší“)

Star glyphs

- Podobné paralelním souřadnicím, ale osy nejsou paralelní, ale **uspořádány do hvězdy**
- Jednotlivé datové položky zase naneseme na osy a propojíme lomennou čárou
- Položka je tedy reprezentována nějakým **uzavřeným polygonem**, nebo uzavřenou lomennou čárou (glyf)
- Neukazujeme jména atributů
- Je lepší, když invertujeme mapování pro „méně je lépe“, protože chceme, aby větší oblast glyfu znamenala, že je to lepší
- Je výhodné je seřadit podle nějaké vlastnosti, abychom se v tom lépe orientovali

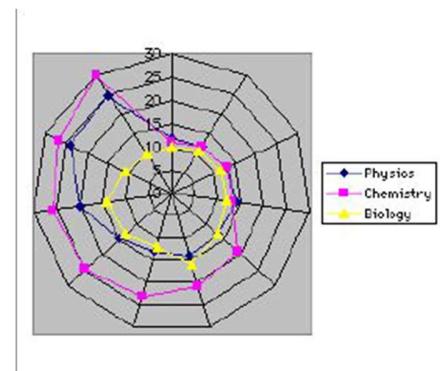


- Lze je dát do scatterplotu podle vybraných atributů, pak vidíme korelaci a můžeme i sledovat, jestli některé glyfy nemají stejný tvar



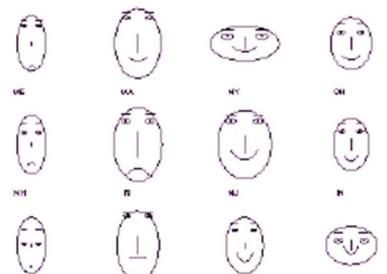
Star plot

- Princip je stejný, ale nebudeme uvažovat star plot jako glyf
- Můžeme zobrazit více položek na osy uspořádané do hvězdice
- Máme méně prostoru na obrazovce pro položky, co jsou blíže k centru plotu
 - o Takže můžeme třeba mapovat data, kterých je více na to, co není blízko centru, abychom měli víc místa pro ně



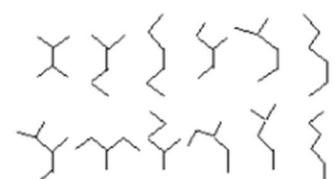
Chernoff faces

- Mapuje odlišné atributy na charakteristiky obličeje člověka
- Používá se schopnost lidí umět detektovat výrazy v obličeji
- Specifické mapování je potřeba:
 - o „dobré“ hodnoty atributů budou mít happy face
 - o „špatné“ hodnoty mají angry face



Stick figures

- Podobné jako star glyfy, ale tvar je trochu jiný
- Atributy jsou mapované na úhly, délky končetin stick figure
- Rozlišné konfigurace končetin napomáhají k rozlišení rozlišných mapování
- Více způsobů, jak vyjádřit mapování atributů



Different configurations

BINNING

- Pro velké množství atributů nebo položek dat
- Dělíme n-dimenzionální prostor na počet binů
- Prostor na obrazovce je rekurzivně rozdělován na bin count pro každou dimenzi
 - o Prostory se dají pak mapovat např. na barvu
 - o Clustery a trendy se ukazují jako nějaký opakující se patterns

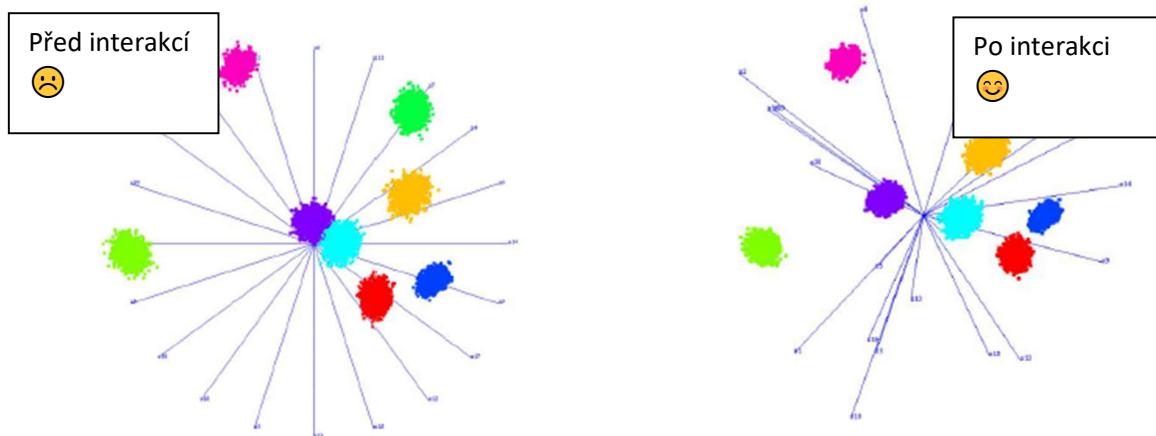
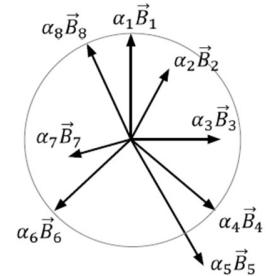
- Dobře škálovatelná technika pro zvyšující se počet dimenzí a položek
- Těžší interpretace

DIMENZIONÁLNÍ PROJEKCE

- Projektuje n-dimenzionální body do 2D prostoru
- Vhodná pro data, co nabývají reálných čísel
- Bázové vektory 2D prostoru jsou nezávislé – více n-dimenzionálních bodů může být projektováno přímo na tuto 2D lokaci
- Např. STAR COORDINATES (nejsou to star glyphs nebo star plot)

STAR COORDINATES

- Osy jsou zase uspořádané do hvězdice
- Úhel mezi osy záleží na počtu dimenzí, který chceme projektovat na 2D (na začátku všechny úhly jsou stejné mezi osy)
- Na počátku můžou být všechny bázové vektory stejně velké
- Polohu bodu ve 2D prostoru spočítáme součtem součinů vektoru x a bázového vektoru B
- Takto ten prostor projektujeme do 2D prostoru, ale deformujeme ho
 - Každá dimenze je namapována na jednotkový vektor
 - Range v každé dimenzi může být odlišný
- Shluky se možná nezformují, nebo se můžou překrývat (kvůli tomu, že vektory nezávislé)

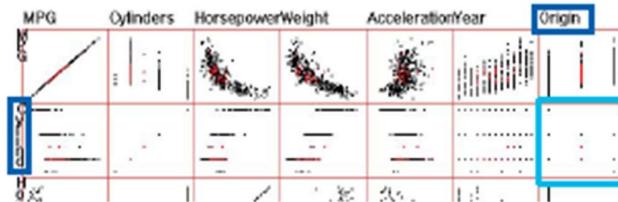


- **Pomáhá interakce** – uživatel by mohl škálovat nebo rotovat bázové vektory
 - Uživatel si může uspořádat osy tak, jak potřebuje, shluky se nemusí pak překrývat
- Pro každý **bázový vektor B zavádíme škálovací faktor/ váhu α**

METODY VHODNÉ PRO DISKRÉTNÍ, NOMINÁLNÍ NEBO ORDINÁLNÍ ATRIBUTY:

Tyto typy dat můžou způsobovat problémy:

- Data dosahují pouze fixního množství „kategorií“ a ty body spadají pod tyto kategorie
- Při použití metod pro spojité data jako např. scatterplot se zobrazí pouze tečky a pro každý nový bod se zobrazí tečka přes tuto tečku. V takovém případě nevidíme vůbec frekvenci těchto teček.
- Špatně se odhaduje kolik položek to je, špatně se odhadují korelace, vzory



Základní principy pro nominální a ordinální atributy

Aggregace

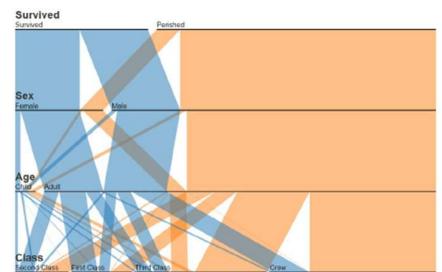
- Pro každý atribut spočítáme počet položek v každý kategorii
- Místo individuálních položek vizualizujeme proporce mezi kategoriemi (e.g. pie chart)

Bargrams (Icicle plot)

- Namapujeme proporce každé kategorie na délku čáry
 - o proporcionálně vůči celkovému počtu
- Nemůžeme sledovat vztahy mezi atributy (kolik dětí přežilo?)
- Např. pohlaví – muž (200), žena (150), děti (100), dospělí (400), ...

Parallel sets

- Vizualizace vztahů mezi atributy
- Bargrams jsou propojené parallelogramy
- Šířka parallelogramu reprezentuje počet položek
- Parallelogramy jsou rozdělené od shora, aby zformovali strukturu stromu



Interakce:

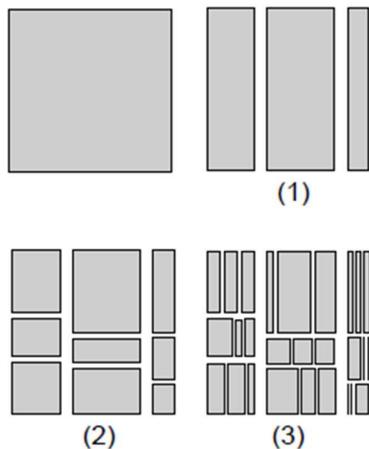
- Předělání řazení kategorií atributu
- Předělání řazení bargramů

Problémy:

- Stromová struktura rozděluje vztahy mezi atributy na atomický level – z toho vzniká visual clutter

Mosaic plot

- Uznávaná metoda pro kategorický proměnný
- Na základě rozdělování 2D prostoru do obdélníků střídavě v ose x a v ose y
- Oblast obdélníku je proporcionalní počtu položek
- Procedura:
 - o Proporcionálně rozděl čtverec dle proměnné X1 na horizontální části podle počtu kategorií
 - o Vezmeme obdélníky, které jsou teď rozdělené na kategorie
 - Obdélník rozdělíme vertikálně, podle počtu kategorií, dle proměnné X2
 - o Takto pokračujeme, dokud všechny proměnné, které jsme chtěli zahrnout jsou zahrnuty
- Je trochu těžší porovnat proporce
- Pouze hierarchické relace mezi atributy jsou obsažené



Mosaic plot matrix

- Lze použít i s faceting
- Zobraz mosaic plot pro každou kombinaci dvou atributů
- Zorganizuj je tak jako scatterplots ve scatter matrixu
- Zase jen závislosti mezi dvěma atributy