

# Técnicas de Visualização para Dados Multivariados

SCC5836/SCC0252 – Visualização Computacional

Profa. Maria Cristina  
*cristina@icmc.usp.br*

Instituto de Ciências Matemáticas e de Computação (ICMC)  
Universidade de São Paulo (USP)

**VICG** Grupo de Visualização,  
 **Imagens e Computação Gráfica**

- 1 Introdução
- 2 Técnicas baseadas em Pontos
  - Scatterplots
  - Projeção Multidimensional
  - Outras Técnicas
- 3 Técnicas baseadas em Linhas
- 4 Técnicas baseadas em Região
- 5 Combinação de Técnicas
- 6 Referências

- Técnicas para **visualização de listas e tabelas de elementos dados (itens)**
- Dados multivariados (multidimensionais): elementos descritos por múltiplos atributos
- Atributos numéricos e/ou categóricos
  - **sem atributo espacial dominante**
  - **sem relação entre os elementos**

- 1 Introdução
- 2 **Técnicas baseadas em Pontos**
  - Scatterplots
  - Projeção Multidimensional
  - Outras Técnicas
- 3 Técnicas baseadas em Linhas
- 4 Técnicas baseadas em Região
- 5 Combinação de Técnicas
- 6 Referências

# Técnicas baseadas em Pontos

- Visualizações baseadas em pontos (*point-based*)
- **Projetam** os itens  $m$ -dimensionais em um **espaço visual**  $p$ -dimensional
  - associam um **marcador gráfico**, ou *glifo*, a cada item
  - cada ponto no espaço  $p$ -dimensional representa um item
- $m$  define a dimensionalidade do conjunto de dados (espaço  $m$ -dimensional)
- terminologia: **variável**, **atributo**, **dimensão**, **campo** muitas vezes tratados como sinônimos

- 1 Introdução
- 2 **Técnicas baseadas em Pontos**
  - **Scatterplots**
  - Projeção Multidimensional
  - Outras Técnicas
- 3 Técnicas baseadas em Linhas
- 4 Técnicas baseadas em Região
- 5 Combinação de Técnicas
- 6 Referências

# Scatterplots ou Matrizes de Scatterplots

- **Scatterplots**: muito populares, mas limitados a dados de duas ou três dimensões.
- Estratégias quando a **dimensionalidade aumenta**

# Scatterplots ou Matrizes de Scatterplots

- **Scatterplots**: muito populares, mas limitados a dados de duas ou três dimensões.
- Estratégias quando a **dimensionalidade aumenta**
  - **Seleção das dimensões**: manualmente, ou por um algoritmo



# Scatterplots ou Matrizes de Scatterplots

- **Scatterplots:** muito populares, mas limitados a dados de duas ou três dimensões.
- Estratégias quando a **dimensionalidade aumenta**
  - **Seleção das dimensões:** manualmente, ou por um algoritmo
  - **Redução das dimensões:** usando técnicas como PCA

# Scatterplots ou Matrizes de Scatterplots

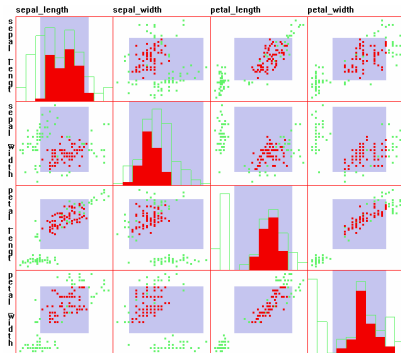
- **Scatterplots:** muito populares, mas limitados a dados de duas ou três dimensões.
- Estratégias quando a **dimensionalidade aumenta**
  - **Seleção das dimensões:** manualmente, ou por um algoritmo
  - **Redução das dimensões:** usando técnicas como PCA
  - **Incorporação das dimensões:** mapear dimensões para outros atributos gráficos (não espaciais)

# Scatterplots ou Matrizes de Scatterplots

- **Scatterplots:** muito populares, mas limitados a dados de duas ou três dimensões.
- Estratégias quando a **dimensionalidade aumenta**
  - **Seleção das dimensões:** manualmente, ou por um algoritmo
  - **Redução das dimensões:** usando técnicas como PCA
  - **Incorporação das dimensões:** mapear dimensões para outros atributos gráficos (não espaciais)
  - **Múltiplos *displays*:** combinar visualizações por meio de superposição ou justaposição (*small multiples*)

# Scatterplots ou Matrizes de Scatterplots

- **Matriz de *scatterplots*** define uma grade de  $m^2$  células ( $m =$  número de dimensões)



**Figura:** Conjunto de dados Iris. Pontos em vermelho foram selecionados pelo usuário.

- 1 Introdução
- 2 **Técnicas baseadas em Pontos**
  - Scatterplots
  - **Projeção Multidimensional**
  - Outras Técnicas
- 3 Técnicas baseadas em Linhas
- 4 Técnicas baseadas em Região
- 5 Combinação de Técnicas
- 6 Referências

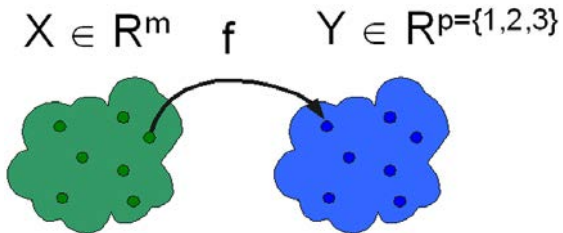
# Projeção Multidimensional

- O princípio dos **métodos de projeção** é **preservar, no espaço projetado  $p$ -dimensional, as relações** existentes entre os elementos no espaço  $m$ -dimensional
  - Por exemplo, relações de **similaridade**, de **vizinhança**, etc.
- Tratamento geométrico do conjunto de dados: itens como pontos (ou vetores) de um espaço  $m$ -dimensional

# Projeção Multidimensional

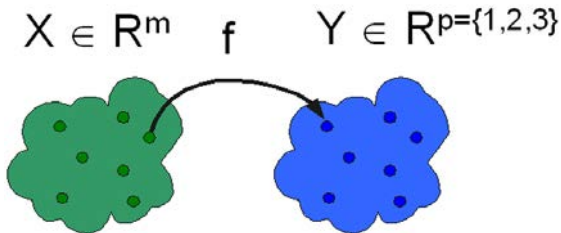
- O princípio dos **métodos de projeção** é **preservar, no espaço projetado  $p$ -dimensional, as relações** existentes entre os elementos no espaço  $m$ -dimensional
    - Por exemplo, relações de **similaridade**, de **vizinhança**, etc.
  - Tratamento geométrico do conjunto de dados: itens como pontos (ou vetores) de um espaço  $m$ -dimensional
- 
- Resultado da projeção é um conjunto de **pontos no plano**
    - Pontos próximos indicam itens relacionados, pontos distantes indicam itens não-relacionados

# Projeção Multidimensional baseada em Distância



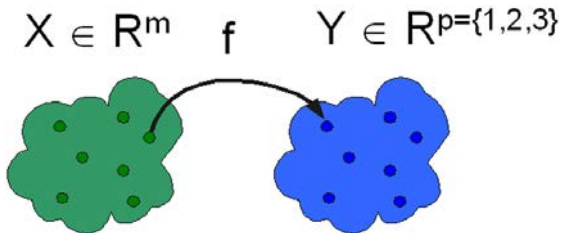


# Projeção Multidimensional baseada em Distância



- $\delta : x_i, x_j \rightarrow \mathbb{R}, x_i, x_j \in X$

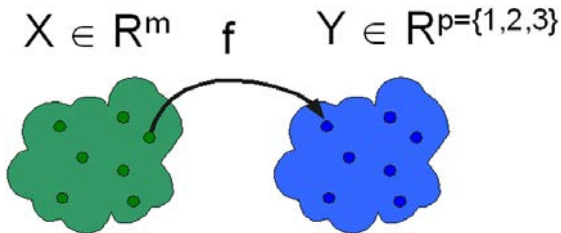
# Projeção Multidimensional baseada em Distância



- $\delta : x_i, x_j \rightarrow \mathbb{R}, x_i, x_j \in X$

- $d : y_i, y_j \rightarrow \mathbb{R}, y_i, y_j \in Y$

# Projeção Multidimensional baseada em Distância



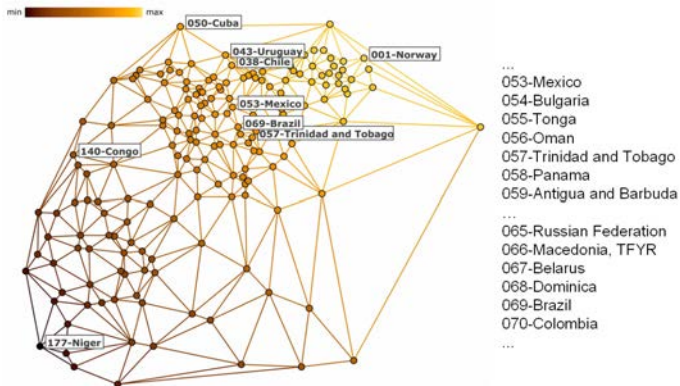
- $\delta : x_i, x_j \rightarrow \mathbb{R}, x_i, x_j \in X$

- $d : y_i, y_j \rightarrow \mathbb{R}, y_i, y_j \in Y$

- $f : X \rightarrow Y, |\delta(x_i, x_j) - d(f(x_i), f(x_j))| \approx 0, \forall x_i, x_j \in X$

# Projeção Multidimensional

- IDH/ONU de 2006 (<http://hdr.undp.org/hdr2006/statistics/>)



# Projeção Multidimensional

Símbolo	Significado
$X$	conjunto de itens no espaço original $m$ -dimensional.
$m$	dimensionalidade do espaço original.
$x_i$	$i$ -ésimo item do espaço original. Quando esse admitir uma representação vetorial, vetor $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{im})$ representa suas coordenadas.
$Y$	conjunto de pontos no espaço projetado $p$ -dimensional.
$p$	dimensionalidade do espaço projetado.
$y_i$	$i$ -ésimo ponto do espaço projetado. Quando esse admitir uma representação vetorial, vetor $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{ip})$ representa suas coordenadas.
$n$	número de itens/pontos (espaço original/projetado).
$\delta(\mathbf{x}_i, \mathbf{x}_j)$	dissimilaridade entre itens $i$ e $j$ no espaço original.
$d(\mathbf{y}_i, \mathbf{y}_j)$	distância entre pontos $i$ e $j$ no espaço projetado.

Tabela: Símbolos mais frequentes e seus significados.

- A maneira de computar a **distância**  $\delta(\mathbf{x}_i, \mathbf{x}_j)$  entre os itens multidimensionais  $\mathbf{X}$  desempenha **papel central**

# Distâncias

- A maneira de computar a **distância**  $\delta(\mathbf{x}_i, \mathbf{x}_j)$  entre os itens multidimensionais  $\mathbf{X}$  desempenha **papel central**

- Distância de *Minkowski* – família de métricas de distância denominadas normas  $L_p$

$$L_p(\mathbf{x}_i, \mathbf{x}_j) = \left( \sum_{k=1}^m |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}} \quad (1)$$

- - $p = 1$ : distância *Manhattan* (*City Block*)
  - $p = 2$ : distância Euclideana
  - $p = \infty$ : distância do infinito  $L_\infty(\mathbf{x}_i, \mathbf{x}_j) = \max_{k=1}^m |x_{ik} - x_{jk}|$

## Propriedades de uma Métrica (Distância)

- 1 **Não-Negatividade:**  $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}, \delta(\mathbf{x}_i, \mathbf{x}_j) \geq 0$
- 2 **Identidade:**  $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}, \mathbf{x}_i = \mathbf{x}_j \Leftrightarrow \delta(\mathbf{x}_i, \mathbf{x}_j) = 0$
- 3 **Simetria:**  $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}, \delta(\mathbf{x}_i, \mathbf{x}_j) = \delta(\mathbf{x}_j, \mathbf{x}_i)$
- 4 **Desigualdade Triangular:**  $\forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in \mathbf{X},$   
 $\delta(\mathbf{x}_i, \mathbf{x}_k) \leq \delta(\mathbf{x}_i, \mathbf{x}_j) + \delta(\mathbf{x}_j, \mathbf{x}_k)$



# Distâncias

- Nem toda dissimilaridade é uma distância (métrica) – não necessariamente satisfaz as propriedades métricas

- Uma **dissimilaridade** pode ser o inverso de uma similaridade  $s(\mathbf{x}_i, \mathbf{x}_j)$

- $\delta(\mathbf{x}_i, \mathbf{x}_j) = e^{-s(\mathbf{x}_i, \mathbf{x}_j)}$

- $\delta(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{s(\mathbf{x}_i, \mathbf{x}_j) + 1}$

- $\delta(\mathbf{x}_i, \mathbf{x}_j) = 1 - s'(\mathbf{x}_i, \mathbf{x}_j)$

- com  $s'(\mathbf{x}_i, \mathbf{x}_j) = \frac{s(\mathbf{x}_i, \mathbf{x}_j) - s_{min}}{s_{max} - s_{min}}$

# Distâncias

- Nem toda dissimilaridade é uma distância (métrica) – não necessariamente satisfaz as propriedades métricas

- Uma **dissimilaridade** pode ser o inverso de uma similaridade  $s(\mathbf{x}_i, \mathbf{x}_j)$

- $\delta(\mathbf{x}_i, \mathbf{x}_j) = e^{-s(\mathbf{x}_i, \mathbf{x}_j)}$

- $\delta(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{s(\mathbf{x}_i, \mathbf{x}_j) + 1}$

- $\delta(\mathbf{x}_i, \mathbf{x}_j) = 1 - s'(\mathbf{x}_i, \mathbf{x}_j)$

- com  $s'(\mathbf{x}_i, \mathbf{x}_j) = \frac{s(\mathbf{x}_i, \mathbf{x}_j) - s_{min}}{s_{max} - s_{min}}$

- Exemplo conhecido: dissimilaridade do cosseno
  - $1 - \cos(\mathbf{x}_i, \mathbf{x}_j)$

- **Categorias** de técnicas de projeção

- **Categorias** de técnicas de projeção
  - Técnicas baseadas em força (*Force-Direct Placement (FDP)*)

- **Categorias** de técnicas de projeção
  - Técnicas baseadas em força (*Force-Direct Placement (FDP)*)
  - Técnicas de decomposição espectral

- **Categorias** de técnicas de projeção
  - Técnicas baseadas em força (*Force-Direct Placement (FDP)*)
  - Técnicas de decomposição espectral
  - Técnicas de redução de dimensionalidade

- **Categorias** de técnicas de projeção
  - Técnicas baseadas em força (*Force-Direct Placement (FDP)*)
  - Técnicas de decomposição espectral
  - Técnicas de redução de dimensionalidade
    - Lineares vs Não Lineares

# Técnicas baseadas em Força

- Técnicas de projeção baseadas no **Modelo de Molas** são as mais simples
  - Busca levar um sistema de objetos conectados por molas a um **estado de equilíbrio** (iterações)
  - Processo iterativo parte de uma configuração inicial
  - Considera as forças geradas pelas molas para aproximar/afastar os objetos, até atingir configuração de equilíbrio
  - **Forças** proporcionais à diferença entre as dissimilaridades  $\delta(\mathbf{x}_i, \mathbf{x}_j)$  e as distâncias  $d(\mathbf{y}_i, \mathbf{y}_j)$



- Os **itens** de dados são modelados como **partículas ponto-massa** conectadas por meio de **molas**
  - Conjunto de partículas sujeito às leis de *Newton*

$$f = m \times a \quad (2)$$

- Os **itens** de dados são modelados como **partículas ponto-massa** conectadas por meio de **molas**
  - Conjunto de partículas sujeito às leis de *Newton*

$$f = m \times a \quad (2)$$

- Resolve-se um sistema de equação diferenciais ordinárias de segunda ordem

$$\begin{cases} v' = a = f/m \\ p' = v \end{cases} \quad (3)$$

# Abordagem de Chalmers

- **Chalmers** apresentou uma **aproximação** do **Modelo de Molas** a fim de acelerar o processamento [Chalmers, 1996]
- Para cada item  $x_i$  cria duas listas
  - $V_i$  armazena as referências aos itens vizinhos de  $x_i$  – mantida do começo ao fim das iterações
  - $S_i$  armazena itens escolhidos aleatoriamente que não estão em  $V_i$  – construída a cada iteração
- Em uma iteração, se a distância de algum elemento de  $S_i$  é menor do que a maior distância para os elementos de  $V_i$ , esse elemento é adicionado à  $V_i$ 
  - As forças são calculadas considerando os elementos presentes em  $S_i$  e  $V_i$

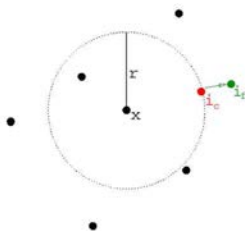
# Modelo Híbrido

- O método de Chalmers foi acelerado com um **Modelo Híbrido** [Morrison et al., 2002]
- Primeiro projeta uma **amostra**  $S$  com  $\sqrt{n}$  itens usando o método de Chalmers
- As posições dos demais itens são **interpoladas**
  - Cada item  $\mathbf{x}_i$  é comparado com a amostra  $S$  para determinar o item  $\mathbf{x}_j$  mais próximo,  $\mathbf{x}_i$  então é mapeado sobre a circunferência de centro  $\mathbf{x}_j$  e raio  $\delta(\mathbf{x}_i, \mathbf{x}_j)$

# Modelo Híbrido

- Realiza uma busca binária para encontrar a posição ideal sobre essa circunferência, minimizando

$$\sum_{z=1}^{n^{1/4}} |\delta(\mathbf{x}_i, \mathbf{x}_z) - d(\mathbf{y}_i, \mathbf{y}_z)|$$



- No fim, calcula um número constante de vetores de força para definir iterativamente a posição final

# Force Scheme

- **Force Scheme** [Tejada et al., 2003] é uma abordagem mais precisa, mas com custo computacional alto,  $O(n^2)$

- 1: **para**  $n=1$  até  $k$  **faça**
- 2:   **para todo**  $y_i \in Y$  **faça**
- 3:     **para todo**  $y_j \in Y$  com  $y_j \neq y_i$  **faça**
- 4:       Calcula  $\vec{v}$  como sendo o vetor de  $y_i$  para  $y_j$
- 5:       Move  $y_j$  na direção de  $\vec{v}$  uma fração de  $\Delta$
- 6:     **fim para**
- 7:   **fim para**
- 8:   Normaliza as coordenadas da projeção no intervalo  $[0, 1]$   
    (ambas as dimensões)
- 9: **fim para**

$$\Delta = \frac{\delta(\mathbf{x}_i, \mathbf{x}_j) - \delta_{min}}{\delta_{max} - \delta_{min}} - d(\mathbf{y}_i, \mathbf{y}_j) \quad (4)$$

# Sammon's Mapping

- A **Sammon's Mapping** [Sammon, 1969] minimiza uma função de perda usando o método de gradientes descendentes

$$S = \frac{1}{\sum_{i < j} \delta(\mathbf{x}_i, \mathbf{x}_j)} \sum_{i < j} \frac{(d(\mathbf{y}_i, \mathbf{y}_j) - \delta(\mathbf{x}_i, \mathbf{x}_j))^2}{\delta(\mathbf{x}_i, \mathbf{x}_j)} \quad (5)$$

- A  $m$ -ésima iteração desse método é definida como

$$y_{pq}(m+1) = y_{pq}(m) - MF \times \Delta_{pq}(m) \quad (6)$$

- Onde  $y_{pq}$  denota a coordenada  $q$  do ponto  $p$  e

$$\Delta_{pq}(m) = \frac{\partial S(m)}{\partial y_{pq}(m)} \bigg/ \left| \frac{\partial^2 S(m)}{\partial y_{pq}^2(m)} \right| \quad (7)$$

# Aproximação de Pekalska

- A técnica **Sammon's Mapping** é  $O(n^3)$ . A fim de reduzir essa complexidade, [Pekalska et al., 1999] definiram uma abordagem em que uma amostra inicial é projetada, e as posições dos demais itens são interpoladas

- Para essa interpolação, encontra o operador  $V$  resolvendo o sistema linear

$$D \times V = Y$$

- onde  $D$  é a matriz de distâncias  $\delta(\mathbf{x}_i, \mathbf{x}_j)$  entre os elementos da amostra e  $Y$  é a projeção da amostra



# Classical Scaling (MDS)

- A **Classical Scaling (MDS)** [Borg and Groenen, 2005] foi uma das primeiras técnicas de projeção (definida em 1930)

- Sejam  $\mathbf{x}_i$  ( $i = 1, \dots, n$ ) as coordenadas de  $n$  pontos em um espaço Euclidiano  $m$ -dimensional, onde  $\mathbf{x}_i = (x_{i1}, \dots, x_{im})^T$ , e seja  $\mathbf{B}$  a matrix do produto interno entre vetores,

$$[\mathbf{B}]_{ij} = b_{ij} = \mathbf{x}_i^T \mathbf{x}_j$$

- Com distância Euclidiana entre os pontos  $i$  e  $j$  dada por

$$\hat{\delta}(\mathbf{x}_i, \mathbf{x}_j)^2 = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j) \quad (8)$$

- A partir de uma matriz da distâncias ( $\hat{\delta}(\mathbf{x}_i, \mathbf{x}_j)^2$ ), encontra a matriz do produto interno  $\mathbf{B}$ , e a partir de  $\mathbf{B}$  calcula as coordenadas dos pontos

# Classical Scaling (MDS)

- A matrix  $\mathbf{B}$  pode ser reescrita como

$$\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{A}\mathbf{H} \quad (9)$$

- $\mathbf{A}$  é a matrix  $[\mathbf{A}]_{ij} = a_{ij} = \hat{\delta}(\mathbf{x}_i, \mathbf{x}_j)^2$
- $\mathbf{H}$  é a matrix de centragem

- Usando decomposição espectral,  $\mathbf{B}$  pode ser escrita em como

$$\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

- Como  $\mathbf{B} = \mathbf{X}\mathbf{X}^T$ , a matrix de coordenadas  $\mathbf{X}$  é dada por

$$\mathbf{X} = \mathbf{V}_1\mathbf{\Lambda}_1^{-\frac{1}{2}}$$

# Isometric Feature Mapping (ISOMAP)

- A **ISOMAP**, ao invés de empregar distâncias Euclidianas entre os itens multidimensionais, emprega **distâncias geodésicas**
  - A **Classical Scaling** é aplicada considerando essas distâncias

## Definição

- Técnicas de **redução de dimensionalidade** buscam encontrar uma representação em um espaço de **dimensionalidade reduzida** que capture o **conteúdo original** dos dados, segundo algum critério

# Principal Component Analysis (PCA)

- **PCA** [Jolliffe, 2002] é uma das técnicas de redução de dimensionalidade mais utilizadas
  - Consegue **capturar** a maior parte da **variabilidade** com poucas dimensões
  - **Elimina** grande parte do “**ruído**” existente, etc.

# Principal Component Analysis (PCA)

- **PCA** [Jolliffe, 2002] é uma das técnicas de redução de dimensionalidade mais utilizadas
  - Consegue **capturar** a maior parte da **variabilidade** com poucas dimensões
  - **Elimina** grande parte do “**ruído**” existente, etc.

- Primeiro cria a matriz de covariância  $\mathbf{C}_{m \times m}$  dos atributos
- Então aplica uma decomposição espectral, encontrando

$$\mathbf{C} = \mathbf{U}\mathbf{A}\mathbf{U}^T \quad (10)$$

- A redução por fim é encontrada fazendo

$$\mathbf{S} = \mathbf{X} \times [u_1, u_2, \dots, u_p] \quad (11)$$

# Least-Square Projection (LSP)

- Na **Least-Square Projection (LSP)** [Paulovich et al., 2008] primeiro é selecionada uma **amostra** dos itens multidimensionais, denominados “pontos de controle”, a qual é projetada em  $\mathbb{R}^p$
- Um sistema linear é construído a partir das **relações de vizinhança** entre os itens no  $\mathbb{R}^m$ , cuja solução dá projeção dos demais itens no **fecho convexo** de seus  $k$  vizinhos mais próximos

# Least-Square Projection (LSP)

- Seja  $V_i = \{p_{i_1}, \dots, p_{i_{k_i}}\}$  um conjunto  $k_i$  pontos em uma vizinhança de um ponto  $p_i$  e sejam  $\tilde{p}_i$  as coordenadas de  $p_i$  no  $\mathbb{R}^d$

- Suponha que  $\tilde{p}_i$  sejam dadas pela seguinte equação

$$\begin{aligned}\tilde{p}_i - \sum_{p_j \in V_i} \alpha_{ij} \tilde{p}_j &= 0 \\ 0 \leq \alpha_{ij} \leq 1; \sum \alpha_{ij} &= 1\end{aligned}\tag{12}$$

- Quando  $\alpha_{ij} = \frac{1}{k_i}$  teremos  $p_i$  no centróide dos pontos em  $V_i$



# Least-Square Projection (LSP)

- A partir dessa equação é definido um conjunto de sistemas lineares

$$L\mathbf{x}_1 = 0, L\mathbf{x}_2 = 0, \dots, L\mathbf{x}_d = 0 \quad (13)$$

- Onde  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d$  são os vetores contendo as coordenadas cartesianas dos pontos e  $L$  é a matriz  $n \times n$  cujas entradas são dadas por

$$l_{ij} = \begin{cases} 1 & i = j \\ -\alpha_{ij} & p_j \in V_i \\ 0 & \text{caso contrário} \end{cases}$$

# Least-Square Projection (LSP)

- Os pontos de controle são inseridos no sistema como novas linhas na matrix
- Assim, dado um conjunto de pontos de controle  $S_c = \{p_{c_1}, \dots, p_{c_{nc}}\}$ , é possível re-escrever o sistema

$$A\mathbf{x} = \mathbf{b} \quad (14)$$

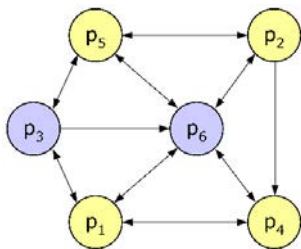
- Onde  $A$  é uma matrix retangular  $(n + nc) \times n$  dada por

$$A = \begin{pmatrix} L \\ C \end{pmatrix}, \quad c_{ij} = \begin{cases} 1 & x_j \text{ é um ponto de controle} \\ 0 & \text{caso contrário} \end{cases}$$

- E  $\mathbf{b}$  é o vetor:

$$b_i = \begin{cases} 0 & i \leq n \\ x_{i_c} & n < i \leq n + nc \end{cases}$$

# Least-Square Projection (LSP)



(a) Grafo de vizinhança.

$$A = \begin{bmatrix} 1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\ 0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\ -1/3 & 0 & 1 & 0 & -1/3 & -1/3 \\ -1/2 & 0 & 0 & 1 & 0 & -1/2 \\ 0 & -1/3 & -1/3 & 0 & 1 & -1/3 \\ -1/4 & -1/4 & 0 & -1/4 & -1/4 & 1 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \leftarrow L \\ \leftarrow C \end{matrix}$$

(b) Matrix Laplaciana.

# Least-Square Projection (LSP)

- O **sistema linear** com os pontos de controle apresenta **rank-completo** e pode ser resolvido aplicando-se **mínimos quadrados**
  - Encontrar  $\mathbf{x}$  que minimize  $\|A\mathbf{x} - \mathbf{b}\|^2$ , isto é,  $\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$

# Piecewise Laplacian Projection (PLP)

- A fim de acelerar o processo da LSP, foi desenvolvida a **Piecewise Laplacian Projection (PLP)** [Paulovich et al., 2011]
- Os dados são sub-divididos em grupos de tamanhos similares, e cada grupo é projetado individualmente com a LSP
  - Cada grupo contém as instâncias mais similares entre si,
  - A coerência espacial entre os grupos é mantida escolhendo pontos de controle de cada grupo, e projetando todos juntos de uma única vez

- **Decomposição espectral**

- Landmarks MDS
- Local Linear Embedding (LLE) [Roweis and Saul, 2000]
- Landmarks ISOMAP
- Pivot MDS
- FASTMAP [Faloutsos and Lin, 1995]

- **Otimização não-linear**

- SMACOF

- **Esquemas baseados em força**

- Glimmer (GPU based) [Ingram et al., 2009]

- **Técnicas Lineares**

- PLMP [Paulovich et al., 2010]
- LAMP

# Exemplos de Projeção

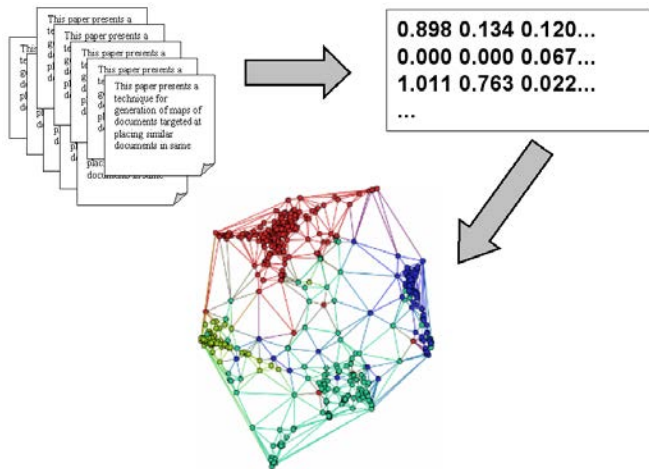


Figura: Processando documentos para visualização (*bag-of-words*).

# Exemplos de Projeção

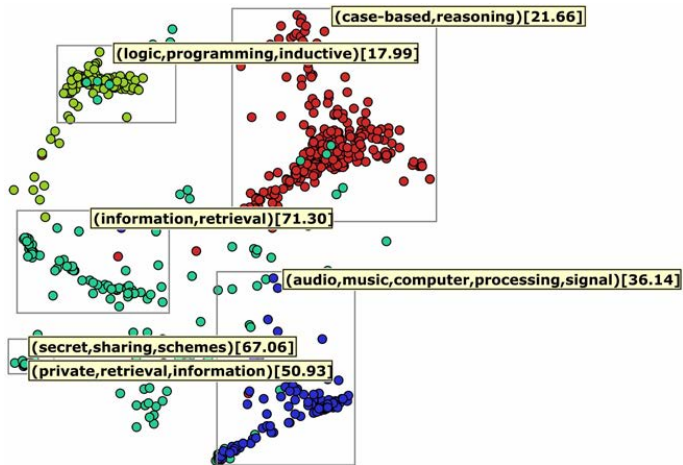


Figura: Projeção de coleção com artigos científicos (LSP).



# Exemplos de Projeção

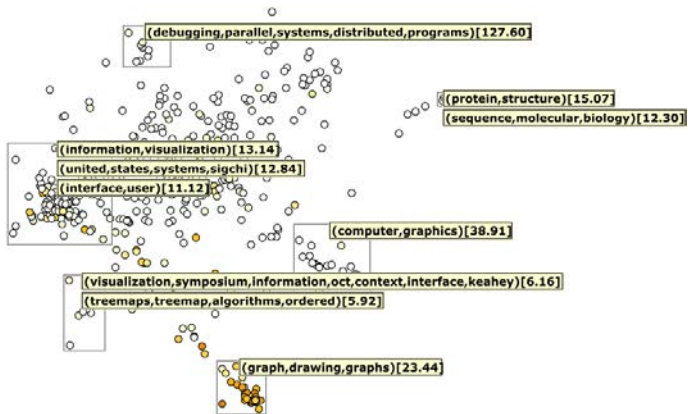


Figura: Projeção de coleção com artigos científicos (LSP).

# Exemplos de Projeção

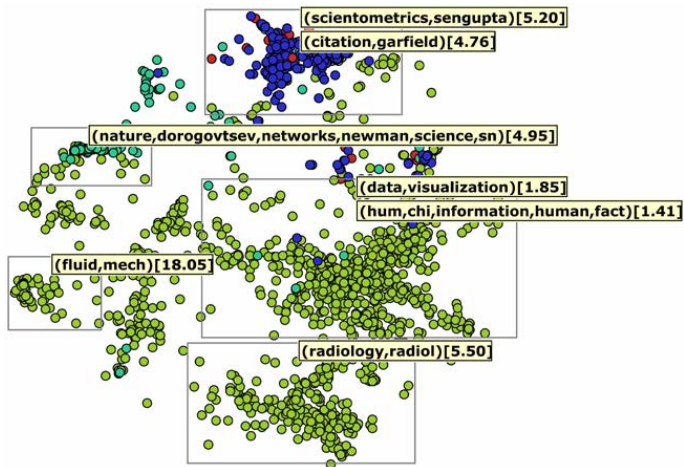


Figura: Projeção de coleção com artigos científicos (LSP).

# Exemplos de Projeção

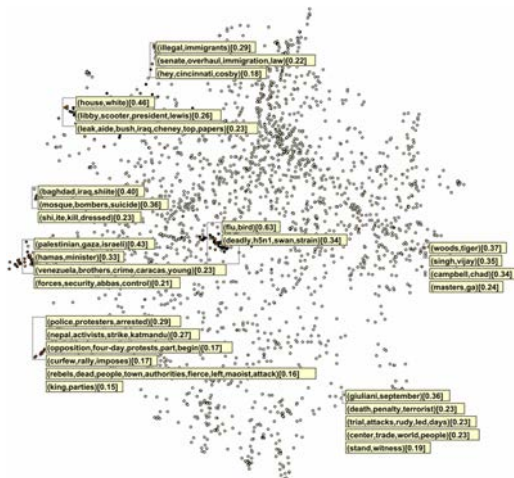


Figura: Projeção de coleção notícias curtas de jornal RSS (LSP).

# Exemplos de Projeção

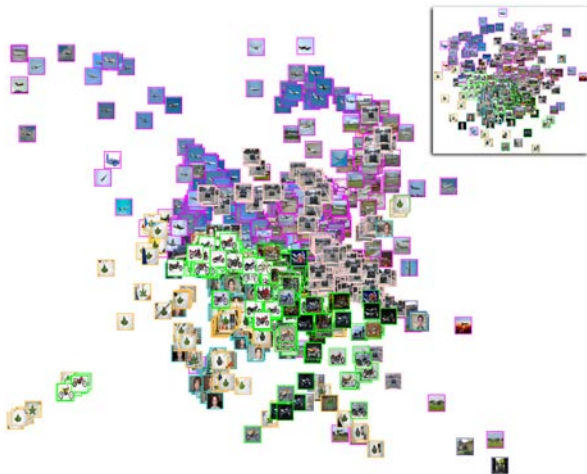


Figura: Projeção de um conjunto de imagens (PLP).

# Exemplos de Projeção

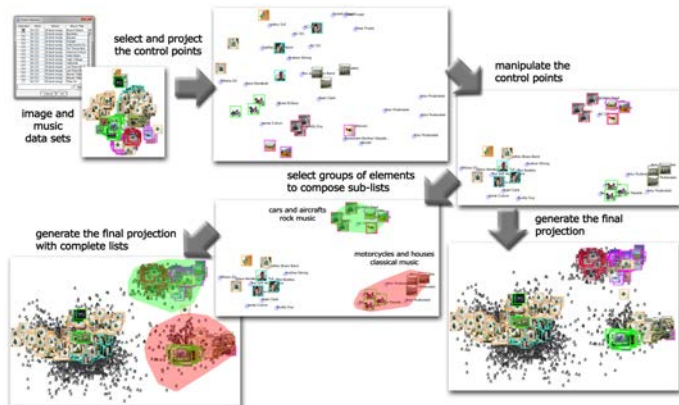


Figura: Projeção para manipulação de dados multimodadais (LAMP)

# Exemplos de Projeção

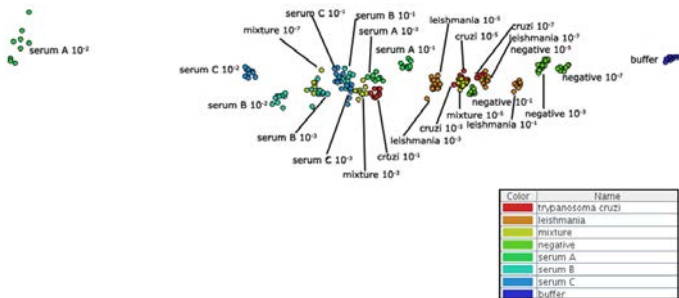


Figura: Projeção de dados de sensores (PCA).

# Exemplos de Projeção

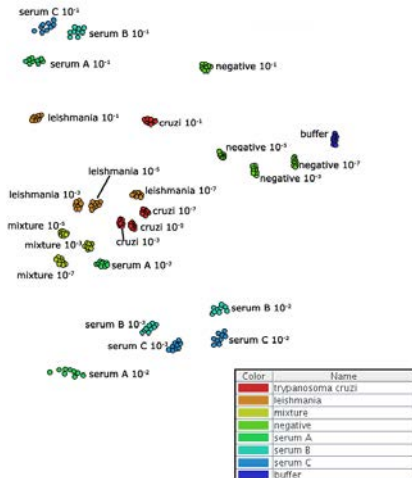
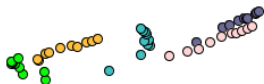


Figura: Projeção de dados de sensores (Force Scheme).

# Exemplos de Projeção



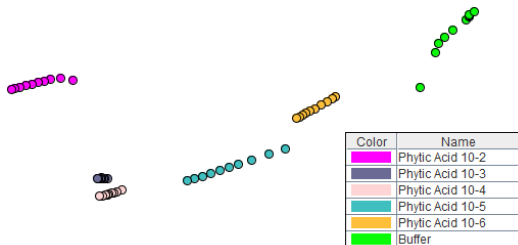
Color	Name
Magenta	Phytic Acid 10-2
Dark Blue	Phytic Acid 10-3
Light Pink	Phytic Acid 10-4
Teal	Phytic Acid 10-5
Yellow	Phytic Acid 10-6
Green	Buffer



**Figura:** Projeção de dados de sensores (Force Scheme) (espectros de medidas). Objetivo é identificar o melhor sensor.

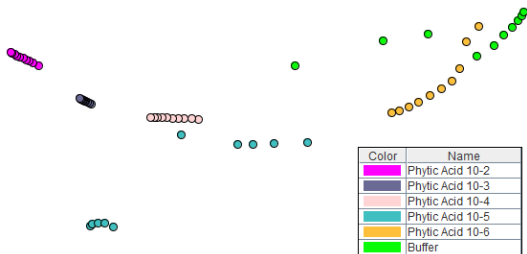


# Exemplos de Projeção



**Figura:** Projeção de dados de sensores (Force Scheme). Objetivo é identificar o melhor sensor.

# Exemplos de Projeção



**Figura:** Projeção de dados de sensores (Force Scheme). Objetivo é identificar o melhor sensor.

# Exemplos de Projeção



**Figura:** Projeção de dados de sensores (Force Scheme). Objetivo é identificar o melhor sensor.

# Exemplos de Projeção

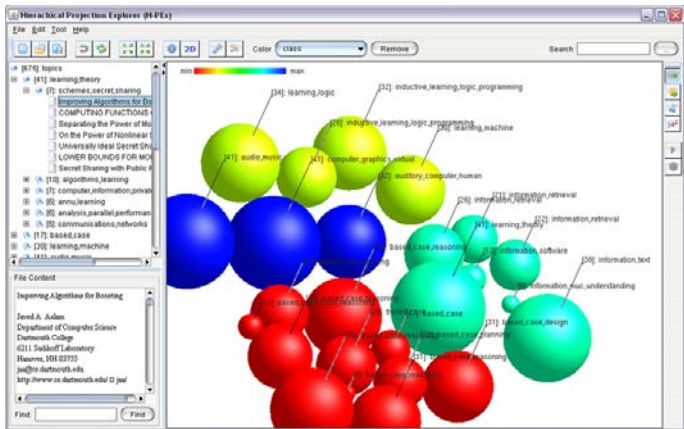


Figura: Hierarchical Point Placement (HiPP) [Paulovich and Minghim, 2008].

- 1 Introdução
- 2 **Técnicas baseadas em Pontos**
  - Scatterplots
  - Projeção Multidimensional
  - **Outras Técnicas**
- 3 Técnicas baseadas em Linhas
- 4 Técnicas baseadas em Região
- 5 Combinação de Técnicas
- 6 Referências

# Técnicas baseadas em Pontos

- Outra técnica baseada em **pontos** muito conhecida é a **RadViz**
  - Para um conjunto  $m$ -dimensional, são definidas  $m$  âncoras, uniformemente posicionadas sobre uma circunferência

# Técnicas baseadas em Pontos

- Outra técnica baseada em **pontos** muito conhecida é a **RadViz**
  - Para um conjunto  $m$ -dimensional, são definidas  $m$  âncoras, uniformemente posicionadas sobre uma circunferência
- Um item de dado é mapeado na posição  $v$  considerando uma circunferência de raio 1 centrada na origem, fazendo
  - Seja  $\mathbf{x} = (x_0, x_1, \dots, x_{m-1})$  o vetor normalizado representando o item e  $A_j$  o vetor normalizado representando a  $j^{\text{ésima}}$  âncora no plano,  $v$  é dado por

$$v = \frac{\sum_{j=0}^{m-1} (A_j x_j)}{\sum_{j=0}^{m-1} x_j}$$

- É importante considerar que **diferentes ordenações** das âncoras levarão a **resultados diferentes**
  - **Interação** pode ajudar o usuário
  - Técnicas que automaticamente **maximizam o espalhamento** dos pontos são boas alternativas



# RadViz

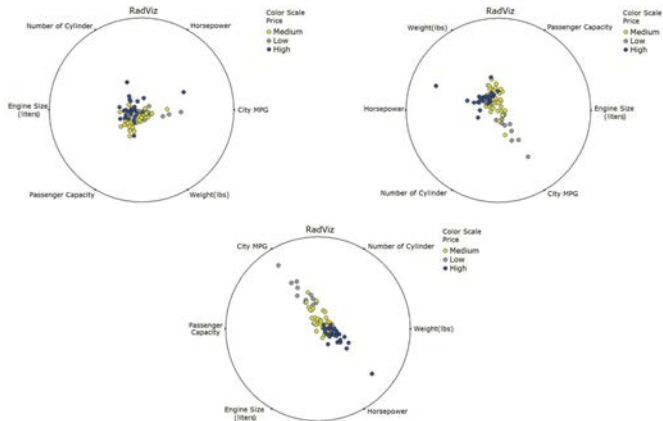


Figura: Diferentes ordenações das âncoras no RadViz. Preço do carro é mapeado para a cor do glifo.

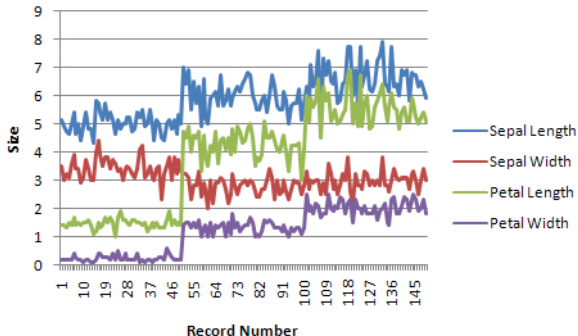
- 1 Introdução
- 2 Técnicas baseadas em Pontos
  - Scatterplots
  - Projeção Multidimensional
  - Outras Técnicas
- 3 Técnicas baseadas em Linhas**
- 4 Técnicas baseadas em Região
- 5 Combinação de Técnicas
- 6 Referências

# Técnicas Baseadas em Linhas

- Nas representações usando **linha**, características dos dados podem ser percebidas por meio de **cruzamentos**, **curvatura**, etc.

# Gráficos de Linhas

- Um gráfico de linha representa no eixo **vertical a faixa de valores** dos itens e o **horizontal alguma ordenação** deles
  - Visualização univariada para **poucas dimensões**



- **Superimposição** empregada com muitas dimensões causa problemas de **sobreposição**

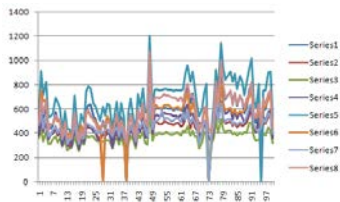
# Gráficos de Linhas

- **Superimposição** empregada com muitas dimensões causa problemas de **sobreposição**
- Possíveis soluções
  - **Empilhar** as linhas usando como base a dimensão anterior (difícil avaliar o valor)

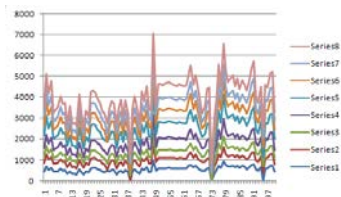
# Gráficos de Linhas

- **Superimposição** empregada com muitas dimensões causa problemas de **sobreposição**
- Possíveis soluções
  - **Empilhar** as linhas usando como base a dimensão anterior (difícil avaliar o valor)
  - **Ordenar** os itens com base em uma dimensão

# Gráficos de Linhas



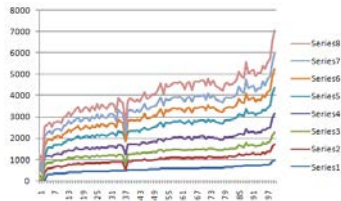
(a)



(b)



(c)



(d)

**Figura:** Gráficos representando salários de professores em 100 universidades. (a) superimposição (b) empilhamento (c) superimposição ordenada (d) empilhamento ordenado.



- Essa técnica só **funciona** se **as unidades das diferentes dimensões forem relacionadas**

# Gráficos de Linhas

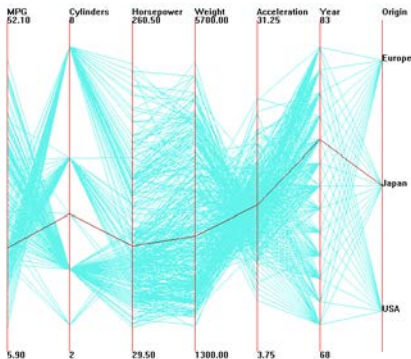
- Essa técnica só **funciona** se **as unidades das diferentes dimensões forem relacionadas**
- Possíveis soluções
  - Pode utilizar **múltiplos eixos verticais** para as diferentes dimensões

# Gráficos de Linhas

- Essa técnica só **funciona** se **as unidades das diferentes dimensões forem relacionadas**
- Possíveis soluções
  - Pode utilizar **múltiplos eixos verticais** para as diferentes dimensões
  - Criar múltiplos **gráficos empilhados coordenados**

# Coordenadas Paralelas

- Na técnica **coordenadas paralelas**, os eixos são paralelos (não ortogonais) espaçados uniformemente
- Os itens são representados por polilinhas, que cruzam esses eixos na posição relativa ao valor da respectiva dimensão/atributo



- A **interpretação** desse tipo de visualização permite

- A **interpretação** desse tipo de visualização permite
  - Localizar **grupos** de polilinhas **semelhantes** (agrupamentos)

- A **interpretação** desse tipo de visualização permite
  - Localizar **grupos** de polilinhas **semelhantes** (agrupamentos)
  - Localizar pontos de **cruzamento** (correlações entre atributos)

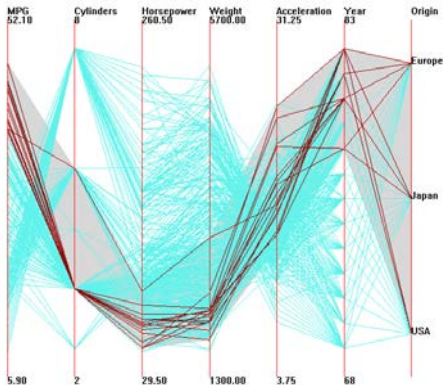
- A **interpretação** desse tipo de visualização permite
  - Localizar **grupos** de polilinhas **semelhantes** (agrupamentos)
  - Localizar pontos de **cruzamento** (correlações entre atributos)
  - Localizar polilinhas **diferentes**



- A **interpretação** desse tipo de visualização permite
  - Localizar **grupos** de polilinhas **semelhantes** (agrupamentos)
  - Localizar pontos de **cruzamento** (correlações entre atributos)
  - Localizar polilinhas **diferentes**
  - Observar a distribuição de valores de cada atributo

# Coordenadas Paralelas

- Pode ser difícil interpretar relações entre **atributos não consecutivos** na representação
  - **Interação** pode ajudar nesta e outras limitações



- Muitas variantes desta técnica

- Muitas variantes desta técnica
  - **Coordenadas paralelas hierárquicas** mostra agrupamentos e não os dados originais

# Coordenadas Paralelas

- Muitas variantes desta técnica
  - **Coordenadas paralelas hierárquicas** mostra agrupamentos e não os dados originais
  - **Semi-transparência** para mostrar grupos em grandes bases de dados

# Coordenadas Paralelas

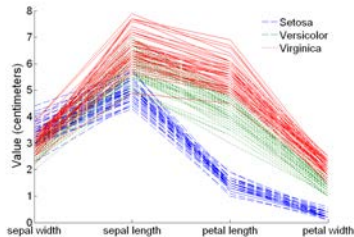
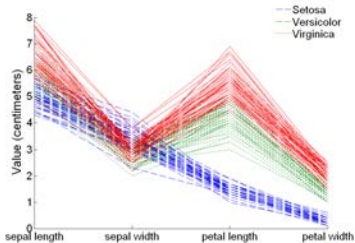
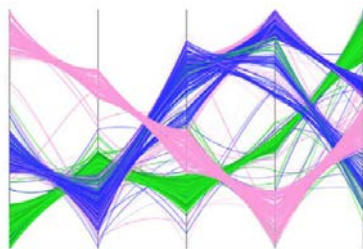
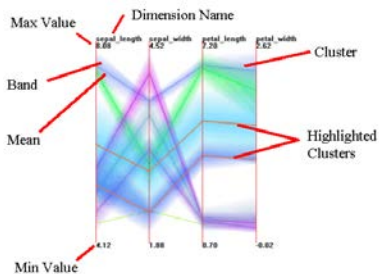
- Muitas variantes desta técnica
  - **Coordenadas paralelas hierárquicas** mostra agrupamentos e não os dados originais
  - **Semi-transparência** para mostrar grupos em grandes bases de dados
  - Agrupamento, re-ordenação e diferentes espaçamentos com base em medidas de **correlação**

- Muitas variantes desta técnica
  - **Coordenadas paralelas hierárquicas** mostra agrupamentos e não os dados originais
  - **Semi-transparência** para mostrar grupos em grandes bases de dados
  - Agrupamento, re-ordenação e diferentes espaçamentos com base em medidas de **correlação**
  - **Incorporação de histogramas** nos eixos para ajudar a interpretação das distribuições

- Muitas variantes desta técnica
  - **Coordenadas paralelas hierárquicas** mostra agrupamentos e não os dados originais
  - **Semi-transparência** para mostrar grupos em grandes bases de dados
  - Agrupamento, re-ordenação e diferentes espaçamentos com base em medidas de **correlação**
  - **Incorporação de histogramas** nos eixos para ajudar a interpretação das distribuições
  - **Curvas nos pontos de intersecção** para melhorar a continuidade entre eixos



# Coordenadas Paralelas



# Curvas de Andrews

- Na técnica de **Curvas de Andrews**, cada item multivariado  $D = (d_1, d_2, \dots, d_N)$  é usado para criar uma curva:

$$f(t) = \frac{d_1}{\sqrt{2}} + d_2 \sin(t) + d_3 \cos(t) + d_4 \sin(2t) + d_5 \cos(2t) + \dots$$

- Com  $-\pi \leq t \leq \pi$

- Para número ímpar de dimensões, o termo final é  $d_N \cos(\frac{N-1}{2}t)$ , para número par é  $d_N \sin(\frac{N}{2}t)$

# Curvas de Andrews

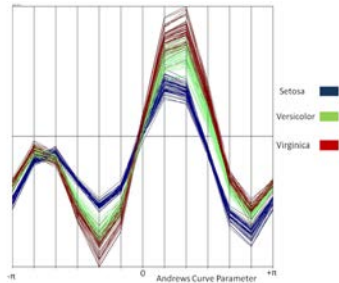
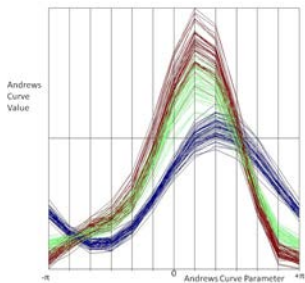
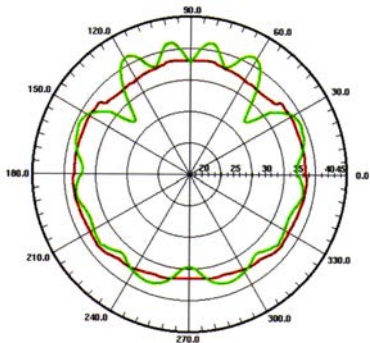


Figura: Curvas de Andrews com diferentes ordenações das dimensões (dados da Iris).

# Técnicas de Eixos Radiais

- Para cada técnica em que os eixos são paralelos, existe uma correspondente em que os eixos são **radiais**
  - P.ex. **Circular line graph**, especialmente útil para eventos cíclicos



# Técnicas de Eixos Radiais

- Para *layouts* que envolvam **mais de um ciclo**, pode-se usar círculos concêntricos, ou um desenho em **espiral**
  - Essa última opção não sofre de problemas de **descontinuidade**

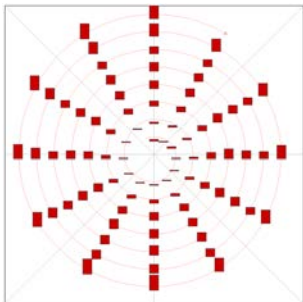


Figura: Exemplo de layout em espiral para um gráfico de barras.

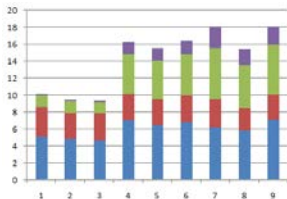
- 1 Introdução
- 2 Técnicas baseadas em Pontos
  - Scatterplots
  - Projeção Multidimensional
  - Outras Técnicas
- 3 Técnicas baseadas em Linhas
- 4 **Técnicas baseadas em Região**
- 5 Combinação de Técnicas
- 6 Referências

# Técnicas baseadas em Região

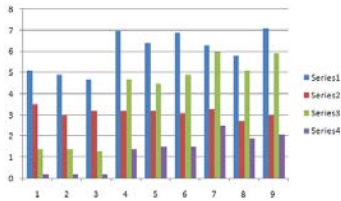
- Técnicas baseadas em **região** apresentam valores empregando **polígonos preenchidos** mapeados para **tamanho, cor, forma** ou outros atributos
  - Apesar das limitações cognitivas muitas técnicas vem sendo desenvolvidas

# Gráficos de Barras e Histogramas

- **Gráficos de barras** são uma visualização muito empregada



(a)



(b)

Figura: Exemplos de gráficos de barras. (a) empilhados (b) agrupados.

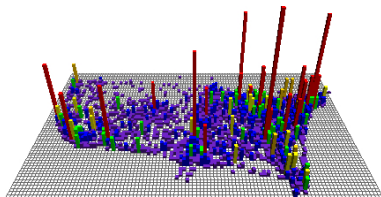
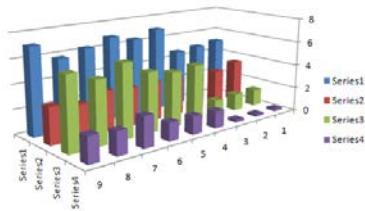


# Gráficos de Barras e Histogramas

- Se a tarefa é estudar a **distribuição dos dados**, **histogramas** são úteis
  - Se atributos têm valores **nominais**, ou variam em um **intervalo pequeno de inteiros** é simples
  - Pode-se ter uma barra para cada valor distinto

# Gráficos de Barras e Histogramas

- Versão 3D do gráfico de barras é conhecido como **Cityscape**
  - Se todas as células forem populadas, também recebe o nome de **Histograma 3D**
  - Técnica bastante popular para dados **geo-referenciados**

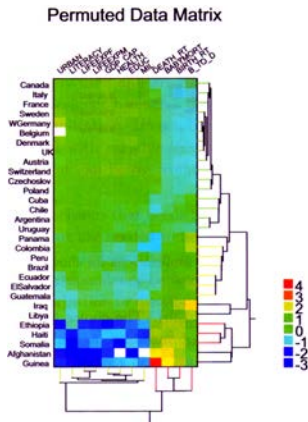


# Displays Tabulares

- Várias visualizações de **dados** organizados como **tabela**

# Displays Tabulares

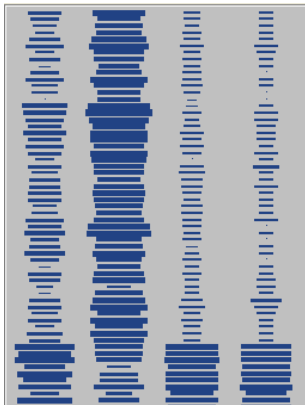
- **Heatmaps** mapeiam os valores em uma tabela de cor e os renderiza como quadrados/retângulos coloridos



- **Permutações e reordenações** de linhas e colunas permitem **melhorar** a visualização
  - Maximizar a diagonal

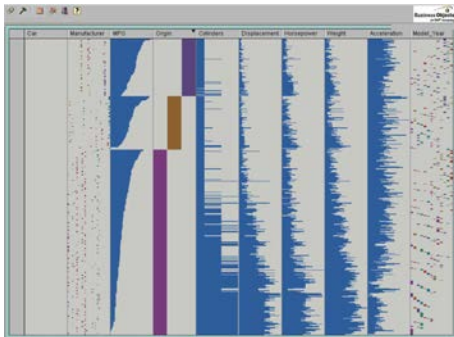
# Displays Tabulares

- **Survey Plots:** varia o **tamanho das células** ao invés de colorir
  - Evita problemas de percepção causados por cores adjacentes



# Displays Tabulares

- **TableLens** combina todas essas ideias e adiciona mecanismos que permitem visualizar toda a tabela, bem como uma **visão detalhada**



- 1 Introdução
- 2 Técnicas baseadas em Pontos
  - Scatterplots
  - Projeção Multidimensional
  - Outras Técnicas
- 3 Técnicas baseadas em Linhas
- 4 Técnicas baseadas em Região
- 5 Combinação de Técnicas**
- 6 Referências



# Combinação de Técnicas

- Existe uma gama de técnicas que **combinam as características** de duas ou mais classes de técnicas apresentadas
  - A criação de glifos

- Diferentes mapeamentos podem ser usados na construção de um **glifo**

- Diferentes mapeamentos podem ser usados na construção de um **glifo**
  - **um-para-um**: cada atributo é mapeado para um canal visual

- Diferentes mapeamentos podem ser usados na construção de um **glifo**
  - **um-para-um**: cada atributo é mapeado para um canal visual
  - **um-para-muitos**: mapeamento redundante para melhorar a precisão

- Diferentes mapeamentos podem ser usados na construção de um **glifo**
  - **um-para-um**: cada atributo é mapeado para um canal visual
  - **um-para-muitos**: mapeamento redundante para melhorar a precisão
  - **muitos-para-um**: diferentes atributos são mapeados para um único canal, separado no espaço, orientação ou outra transformação

- Alguns exemplos de **glifos** são

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro



- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios
  - **Stick figures** – tamanho, ângulo, cor do membro

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios
  - **Stick figures** – tamanho, ângulo, cor do membro
  - **Trees** – tamanho, espessura, ângulos dos ramos, etc.

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios
  - **Stick figures** – tamanho, ângulo, cor do membro
  - **Trees** – tamanho, espessura, ângulos dos ramos, etc.
  - **Autoglifo** – cor de caixas

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios
  - **Stick figures** – tamanho, ângulo, cor do membro
  - **Trees** – tamanho, espessura, ângulos dos ramos, etc.
  - **Autoglifo** – cor de caixas
  - **Boxes** – altura, comprimento, profundidade, etc.

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios
  - **Stick figures** – tamanho, ângulo, cor do membro
  - **Trees** – tamanho, espessura, ângulos dos ramos, etc.
  - **Autoglifo** – cor de caixas
  - **Boxes** – altura, comprimento, profundidade, etc.
  - **Hedgehogs** – picos em um campo vetorial, com variação em orientação grossura

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios
  - **Stick figures** – tamanho, ângulo, cor do membro
  - **Trees** – tamanho, espessura, ângulos dos ramos, etc.
  - **Autoglifo** – cor de caixas
  - **Boxes** – altura, comprimento, profundidade, etc.
  - **Hedgehogs** – picos em um campo vetorial, com variação em orientação grossura
  - **Faces** – tamanho e posição dos olhos, nariz, boca, etc.

- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios
  - **Stick figures** – tamanho, ângulo, cor do membro
  - **Trees** – tamanho, espessura, ângulos dos ramos, etc.
  - **Autoglifo** – cor de caixas
  - **Boxes** – altura, comprimento, profundidade, etc.
  - **Hedgehogs** – picos em um campo vetorial, com variação em orientação grossura
  - **Faces** – tamanho e posição dos olhos, nariz, boca, etc.
  - **Arrows** – tamanho, comprimento, cor



- Alguns exemplos de **glifos** são
  - **Profiles** – altura e cor de barras
  - **Stars** – tamanho de raios igualmente espaçados emanando do centro
  - **Anderson/metroglifos** – tamanho de raios
  - **Stick figures** – tamanho, ângulo, cor do membro
  - **Trees** – tamanho, espessura, ângulos dos ramos, etc.
  - **Autoglifo** – cor de caixas
  - **Boxes** – altura, comprimento, profundidade, etc.
  - **Hedgehogs** – picos em um campo vetorial, com variação em orientação grossura
  - **Faces** – tamanho e posição dos olhos, nariz, boca, etc.
  - **Arrows** – tamanho, comprimento, cor
  - etc.

# Glyphs e Ícones



**PROFILE GLYPHS**

**STARS AND  
METROGLYPHS**



**STICKS AND TREES**



**AUTOGLYPH/BOX GLYPH**



**FACE GLYPHS**



**ARROWS/WEATHERVANES**

Figura: Exemplos de glifos multivariados.

- Na **escolha do glifo**, deve-se considerar:

- Na **escolha do glifo**, deve-se considerar:
  - **Viés imposto** pelo atributo gráfico usado (linha pode ser melhor que orientação)

- Na **escolha do glifo**, deve-se considerar:
  - **Viés imposto** pelo atributo gráfico usado (linha pode ser melhor que orientação)
  - **Glifos adjacentes** serão mais **facilmente comparados** dos que os afastados

- Na **escolha do glifo**, deve-se considerar:
  - **Viés imposto** pelo atributo gráfico usado (linha pode ser melhor que orientação)
  - **Glifos adjacentes** serão mais **facilmente comparados** dos que os afastados
  
- Uma vez escolhido o glifo, existem  $N!$  **diferentes possibilidades** de mapeamento das dimensões
  - Qual ordem escolher no mapeamento

- Algumas **estratégias** para escolher o **mapeamento**

- Algumas **estratégias** para escolher o **mapeamento**
  - **Dimensões correlacionadas** podem ser mapeadas para canais visuais **adjacentes** – ajuda a revelar padrões e outliers



- Algumas **estratégias** para escolher o **mapeamento**
  - **Dimensões correlacionadas** podem ser mapeadas para canais visuais **adjacentes** – ajuda a revelar padrões e outliers
  - Dimensões mapeadas de forma a compor **glifos simétricos** – glifos não simétricos serão facilmente reconhecidos

- Algumas **estratégias** para escolher o **mapeamento**
  - **Dimensões correlacionadas** podem ser mapeadas para canais visuais **adjacentes** – ajuda a revelar padrões e outliers
  - Dimensões mapeadas de forma a compor **glifos simétricos** – glifos não simétricos serão facilmente reconhecidos
  - Dimensões podem ser mapeadas de **acordo com seus valores**

- Algumas **estratégias** para escolher o **mapeamento**
  - **Dimensões correlacionadas** podem ser mapeadas para canais visuais **adjacentes** – ajuda a revelar padrões e outliers
  - Dimensões mapeadas de forma a compor **glifos simétricos** – glifos não simétricos serão facilmente reconhecidos
  - Dimensões podem ser mapeadas de **acordo com seus valores**
  - Dimensões podem ser mapeadas de acordo com o **conhecimento do usuário**

- Três **estratégias de layout** (posicionamento)

- Três **estratégias de layout** (posicionamento)
  - **uniforme**: glifos escalados e posicionados com espaçamento uniforme para ocupar a tela inteira – evita sobreposições

- Três **estratégias de layout** (posicionamento)
  - **uniforme**: glifos escalados e posicionados com espaçamento uniforme para ocupar a tela inteira – evita sobreposições
  - **dirigido-a-dados**: posição dos glifos determinada pelos valores de dados – podem ser posicionados usando scatterplots ou projeções multidimensionais

- Três **estratégias de layout** (posicionamento)
  - **uniforme**: glifos escalados e posicionados com espaçamento uniforme para ocupar a tela inteira – evita sobreposições
  - **dirigido-a-dados**: posição dos glifos determinada pelos valores de dados – podem ser posicionados usando scatterplots ou projeções multidimensionais
  - **dirigido-a-estrutura**: considera, se existir, alguma estrutura implícita, p.ex. cíclica ou hierárquica

# Glifos e Ícones

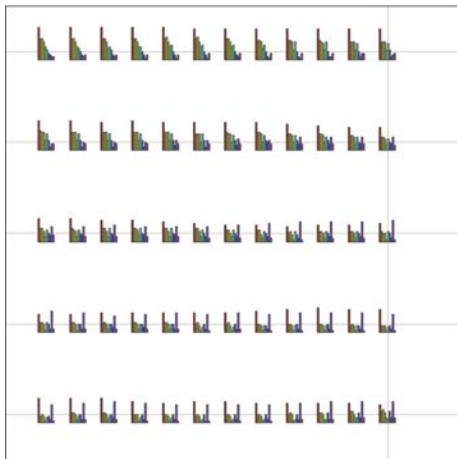
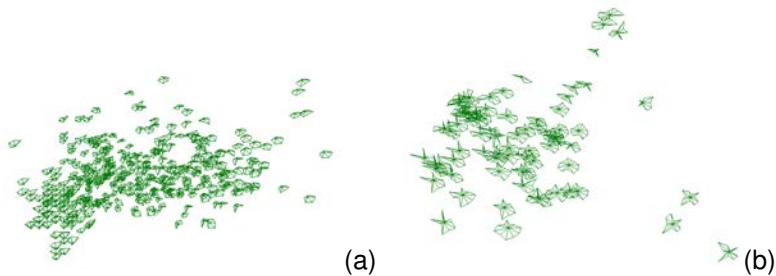


Figura: Exemplos de posicionamento uniforme.

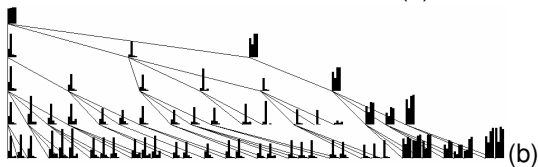
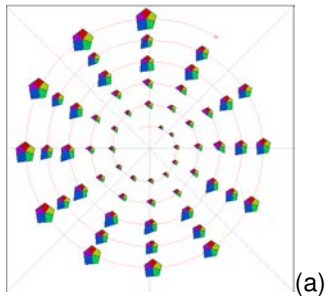


# Glifos e Ícones



**Figura:** Exemplos de posicionamento dirigido-a-dados. (a) usando dimensões (b) usando PCA.

# Glifos e Ícones



**Figura:** Exemplos de posicionamento dirigido-a-estrutura. (a) estrutura em espiral (b) estrutura hierárquica.

# Display Denso de Pixels

- **Display Denso de Pixels** (técnicas orientadas-a-pixel) mapeiam cada **valor** para **pixels individuais** e criam um polígono preenchido para representar cada dimensão dos dados
  - Usam todo o espaço de tela
  - Cada **valor** de dado define a **cor do pixel**

## Display Denso de Pixels

- Na forma mais simples, cada **dimensão** irá gerar uma **imagem no display**

# Display Denso de Pixels

- Na forma mais simples, cada **dimensão** irá gerar uma **imagem no display**
- Elementos precisam ser desenhados de modo que **itens próximos fiquem próximos**

# Display Denso de Pixels

- Na forma mais simples, cada **dimensão** irá gerar uma **imagem no display**
- Elementos precisam ser desenhados de modo que **itens próximos fiquem próximos**
  - Pode-se criar uma imagem fazendo uma travessia da esquerda-para-direita da **direita-para-esquerda**

# Display Denso de Pixels

- Na forma mais simples, cada **dimensão** irá gerar uma **imagem no display**
- Elementos precisam ser desenhados de modo que **itens próximos fiquem próximos**
  - Pode-se criar uma imagem fazendo uma travessia da esquerda-para-direita da **direita-para-esquerda**
  - Pode-se adotar uma travessia em **espiral**

# Display Denso de Pixels

- Na forma mais simples, cada **dimensão** irá gerar uma **imagem no display**
- Elementos precisam ser desenhados de modo que **itens próximos fiquem próximos**
  - Pode-se criar uma imagem fazendo uma travessia da esquerda-para-direita da **direita-para-esquerda**
  - Pode-se adotar uma travessia em **espiral**
  - Pode-se adotar uma **curva recursiva** para preencher o layout (pontos próximos na lista ficam próximos no display)



# Display Denso de Pixels

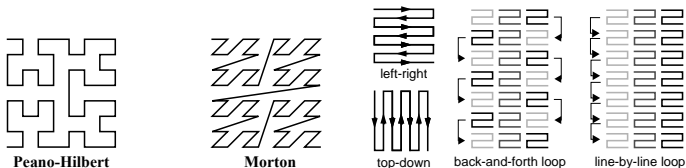
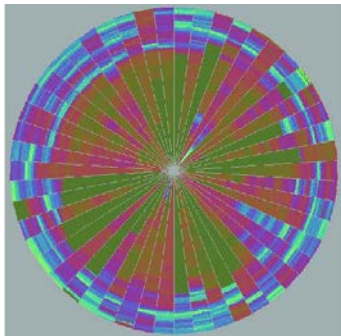
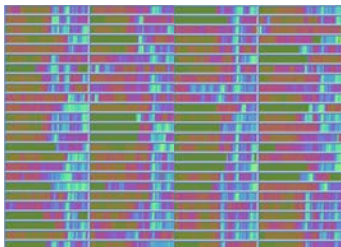


Figura: Exemplos de padrões de layout para displays densos de pixels.

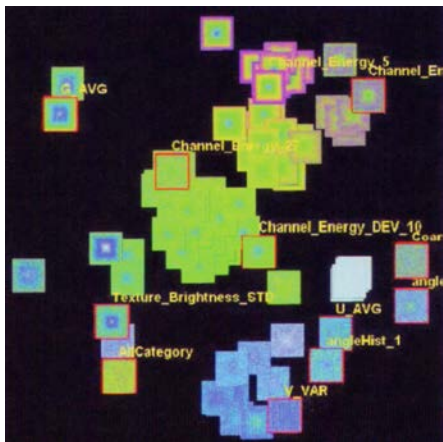
# Display Denso de Pixels

- As **imagens** de cada dimensão podem ser **posicionadas na tela** de várias maneiras, p.ex. uma **grade** ou em **segmentos de círculo**



# Display Denso de Pixels

- Outra possibilidade é **usar um método de projeção** para posicionar as dimensões similares proximamente



## Display Denso de Pixels

- A **ordenação** dos dados (quando possível) **pode ajudar** a identificar padrões interessantes

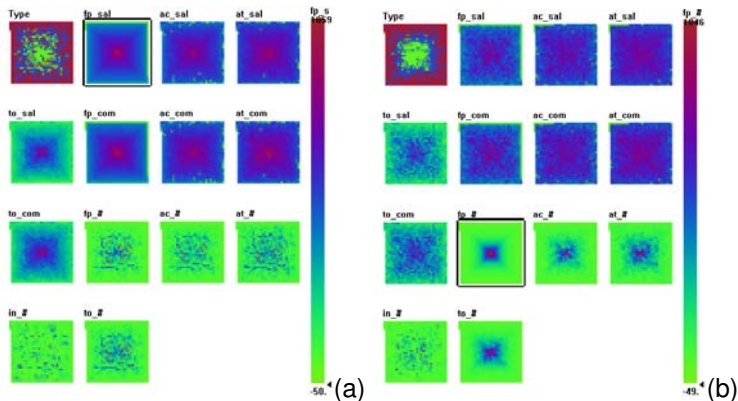
# Display Denso de Pixels

- A **ordenação** dos dados (quando possível) **pode ajudar** a identificar padrões interessantes
  - **Ordenar** com base em uma **dimensão** ajudar a revelar agrupamentos dentro dessa dimensão

# Display Denso de Pixels

- A **ordenação** dos dados (quando possível) **pode ajudar** a identificar padrões interessantes
  - **Ordenar** com base em uma **dimensão** ajudar a revelar agrupamentos dentro dessa dimensão
  - **Ordenar** com base na **distância** a um vetor  $n$ -dimensional ajuda a revelar agrupamentos em várias dimensões

# Display Denso de Pixels



**Figura:** Exemplos de diferentes ordenações. (a) dados ordenados com base no salário dos professores: pouca correlação com outros atributos. (b) dados ordenados com base em número de “full professors”: somente relacionado com o número total de empregados.

- 1 Introdução
- 2 Técnicas baseadas em Pontos
  - Scatterplots
  - Projeção Multidimensional
  - Outras Técnicas
- 3 Técnicas baseadas em Linhas
- 4 Técnicas baseadas em Região
- 5 Combinação de Técnicas
- 6 Referências



- Ward, M., Grinstein, G. G., Keim, D. **Interactive data visualization foundations, techniques, and applications.** Natick, Mass., A K Peters, 2010.



Borg, I. and Groenen, P. J. F. (2005).

*Modern Multidimensional Scaling: Theory and Applications.*  
Springer, second edition.



Chalmers, M. (1996).

A linear iteration time layout algorithm for visualising high-dimensional data.  
In *VIS '96: Proceedings of the 7th conference on Visualization '96*, pages 127–ff., Los Alamitos, CA, USA. IEEE Computer Society Press.



Faloutsos, C. and Lin, K. (1995).

FastMap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia databases.  
In *ACM SIGMOD*, pages 163–174.



Ingram, S., Munzner, T., and Olano, M. (2009).

Glimmer: Multilevel MDS on the GPU.  
*IEEE Trans. Vis. Comp. Graph.*, 15(2):249–261.

# Referências II



Jolliffe, I. T. (2002).  
*Principal Component Analysis*.  
Springer.



Morrison, A., Ross, G., and Chalmers, M. (2002).  
A hybrid layout algorithm for sub-quadratic multidimensional scaling.  
In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 152, Washington, DC, USA. IEEE Computer Society.



Paulovich, F., Eler, D., Poco, J., Botha, C., Minghim, R., and Nonato, L. (2011).  
Piecewise laplacian-based projection for interactive large data exploration and organization.  
*Computer Graphics Forum*, 30(3):1091–1100.



Paulovich, F. V. and Minghim, R. (2008).  
HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections.  
*IEEE Trans. Visual. Comp. Graph.*, 14(6):1229–1236.



Paulovich, F. V., Nonato, L. G., Minghim, R., and Levkowitz, H. (2008).  
Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping.  
*IEEE Trans. Visual. Comp. Graph.*, 14(3):564–575.



Paulovich, F. V., Silva, C. T., and Nonato, L. G. (2010).  
Two-phase mapping for projecting massive data sets.  
*IEEE Trans. on Vis. Comp. Graph.*, 16(6):1281–1290.

# Referências III



Pekalska, E., de Ridder, D., Duin, R. P. W., and Kraaijveld, M. A. (1999).

A new method of generalizing Sammon mapping with application to algorithm speed-up. In Boasson, M., Kaandorp, J. A., Tonino, J. F. M., and Vosselman, M. G., editors, *Annual Conf. Advanced School for Comput. Imag.*, pages 221–228.



Roweis, S. T. and Saul, L. K. (2000).

Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.



Sammon, J. W. (1969).

A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, 18(5):401–409.



Tejada, E., Minghim, R., and Nonato, L. G. (2003).

On improved projection techniques to support visual exploration of multidimensional data sets.

*Information Visualization*, 2(4):218–231.