Aprendizado Não-Supervisionado

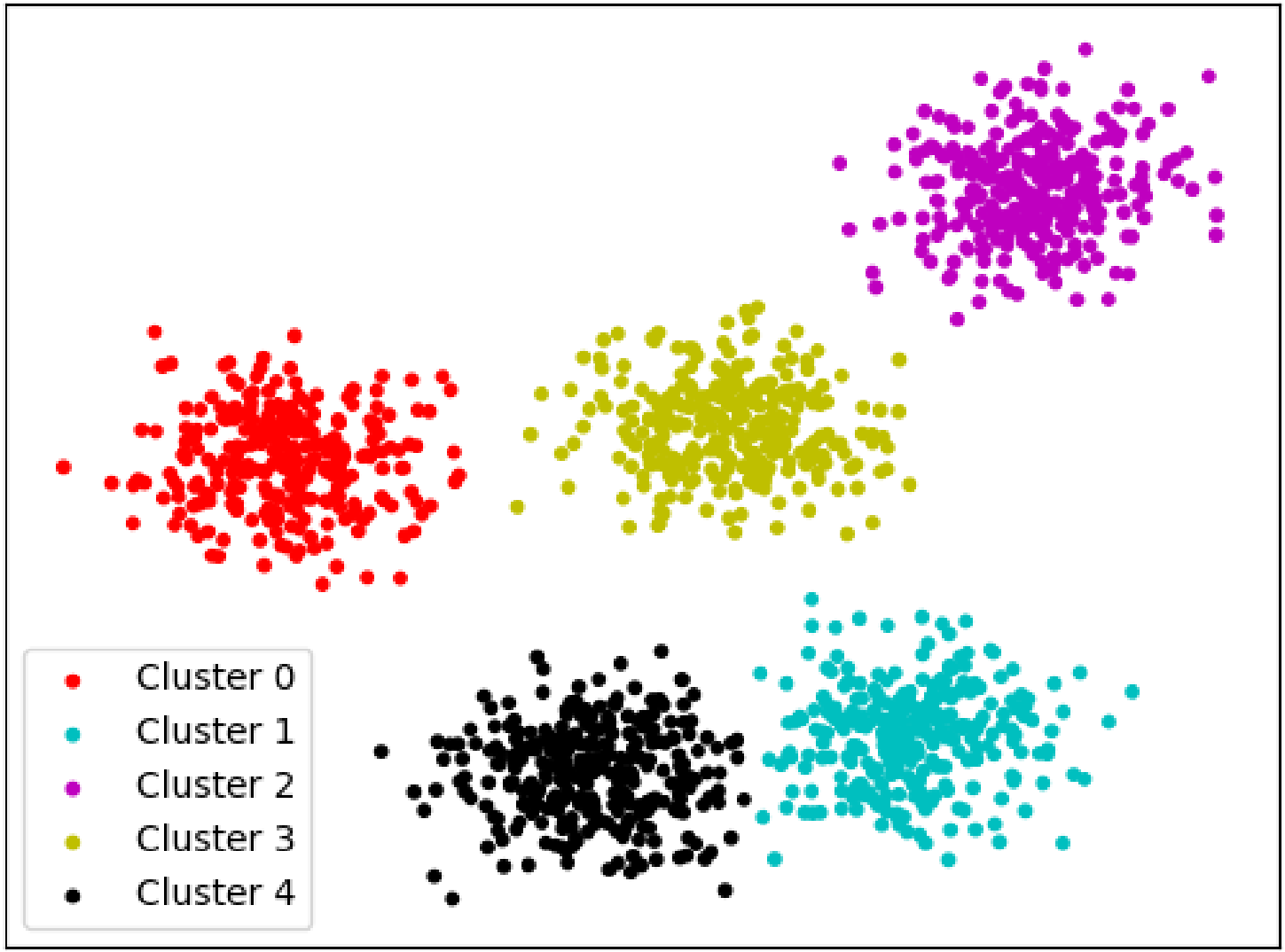
Prof. Danilo Silva

EEL7514/EEL7513 - Tópico Avançado em Processamento de Sinais:
Introdução ao Aprendizado de Máquina

EEL / CTC / UFSC

Clustering com K-means

Clustering



Clustering

- Conjunto de dados não-rotulados: $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^n$
- Problema: separar os dados em K grupos (clusters) de amostras "similares", i.e., que estejam mais "próximas" das amostras do mesmo grupo do que das de outros grupos
- Clustering baseado em centróides (k-*means*): determinar K centróides $\boldsymbol{\mu}_k$ e atribuir cada amostra ao centróide mais próximo
- Outros algoritmos de clustering:
  - Clustering hierárquico/aglomerativo
  - Clustering baseado em distribuição de probabilidade
  - Clustering baseado em densidade de pontos
  - Clustering baseado em grafos

Notação

- $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^n$: amostras/pontos
- K: número de clusters (escolhido a priori)
- $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^n$: médias/centróides (representantes dos clusters)
- $c^{(i)} \in \{1, \ldots, K\}$: índice do cluster ao qual a amostra $\mathbf{x}^{(i)}$ está atribuída
- $\mathcal{S}_k = \{\mathbf{x}^{(i)} : c^{(i)} = k\}$: k-ésimo cluster
- Função custo:

$$J(c^{(1)}, \ldots, c^{(m)}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K) = \sum_{i=1}^{m} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}}\|^2$$

$$= \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{S}_k} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2$$

Otimização Alternada

- Se os centróides μ_k estão fixos, a solução ótima da atribuição é

$$c^{(i)} = \underset{k}{\operatorname{argmin}} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|^2$$

  isto é, atribui-se $\mathbf{x}^{(i)}$ ao cluster cujo centróide esteja mais próximo

- Se as atribuições $c^{(i)}$ estão fixas, a solução ótima para $\boldsymbol{\mu}_k$ é

$$\boldsymbol{\mu}_k = \frac{1}{|\mathcal{S}_k|} \sum_{\mathbf{x} \in \mathcal{S}_k} \mathbf{x}$$

  isto é, escolhe-se $\boldsymbol{\mu}_k$ como sendo a média (centróide) das amostras pertencentes ao cluster k

- Alternar estas otimizações nunca pode aumentar o custo

Algoritmo k-means

- Inicialize aleatoriamente $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^n$
- Repita até a convergência:
  - Para $i = 1, \ldots, m$:

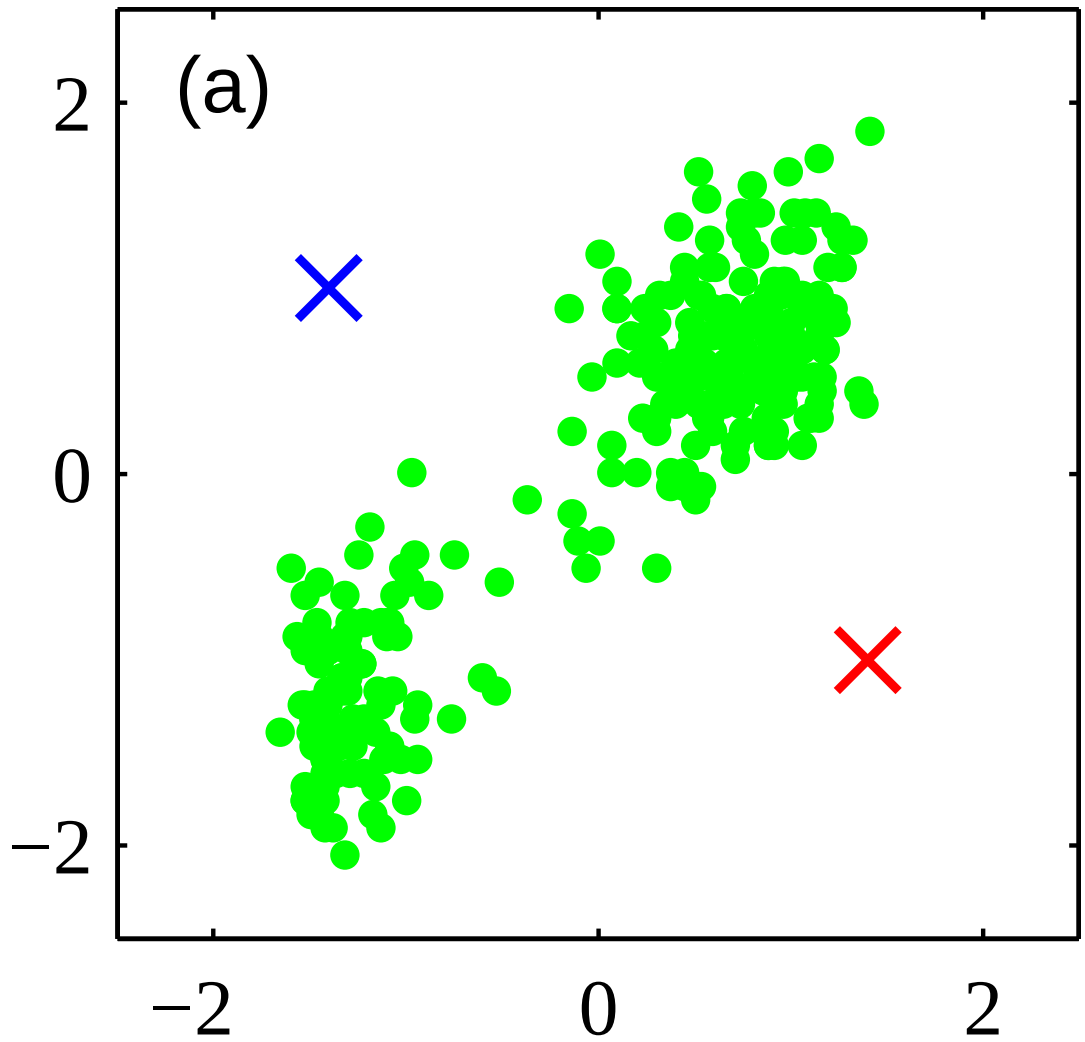
$$c^{(i)} = \operatorname*{argmin}_k \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|^2$$

  - Para $k = 1, \ldots, K$:

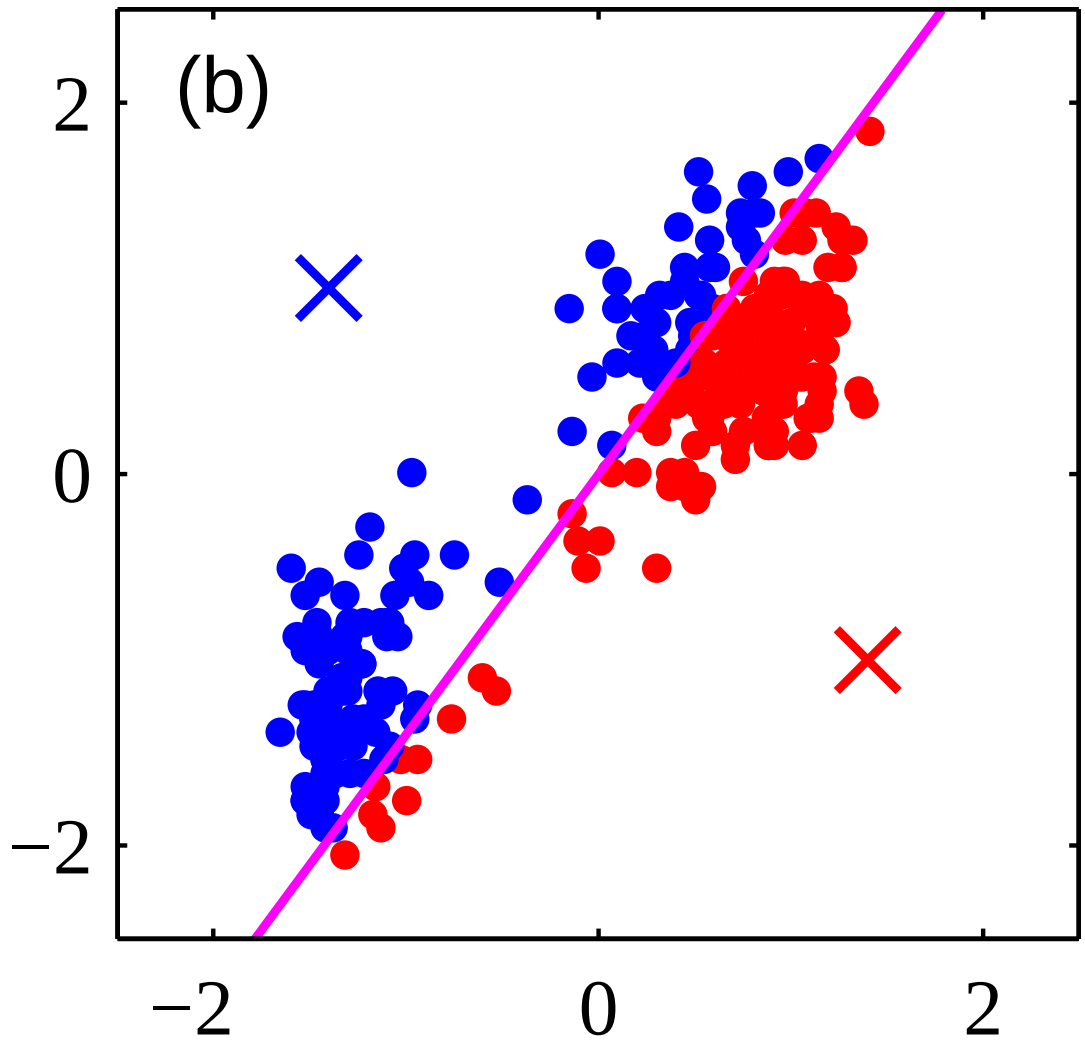
$$\boldsymbol{\mu}_k = \frac{1}{|\{i : c^{(i)} = k\}|} \sum_{i:c^{(i)}=k} \mathbf{x}^{(i)}$$

- Obs: o algoritmo sempre converge, mas não necessariamente para o ótimo global
- Normalmente utilizado com múltiplas reinicializações
  - Escolhe-se a melhor de N tentativas (ex: $N = 100$)

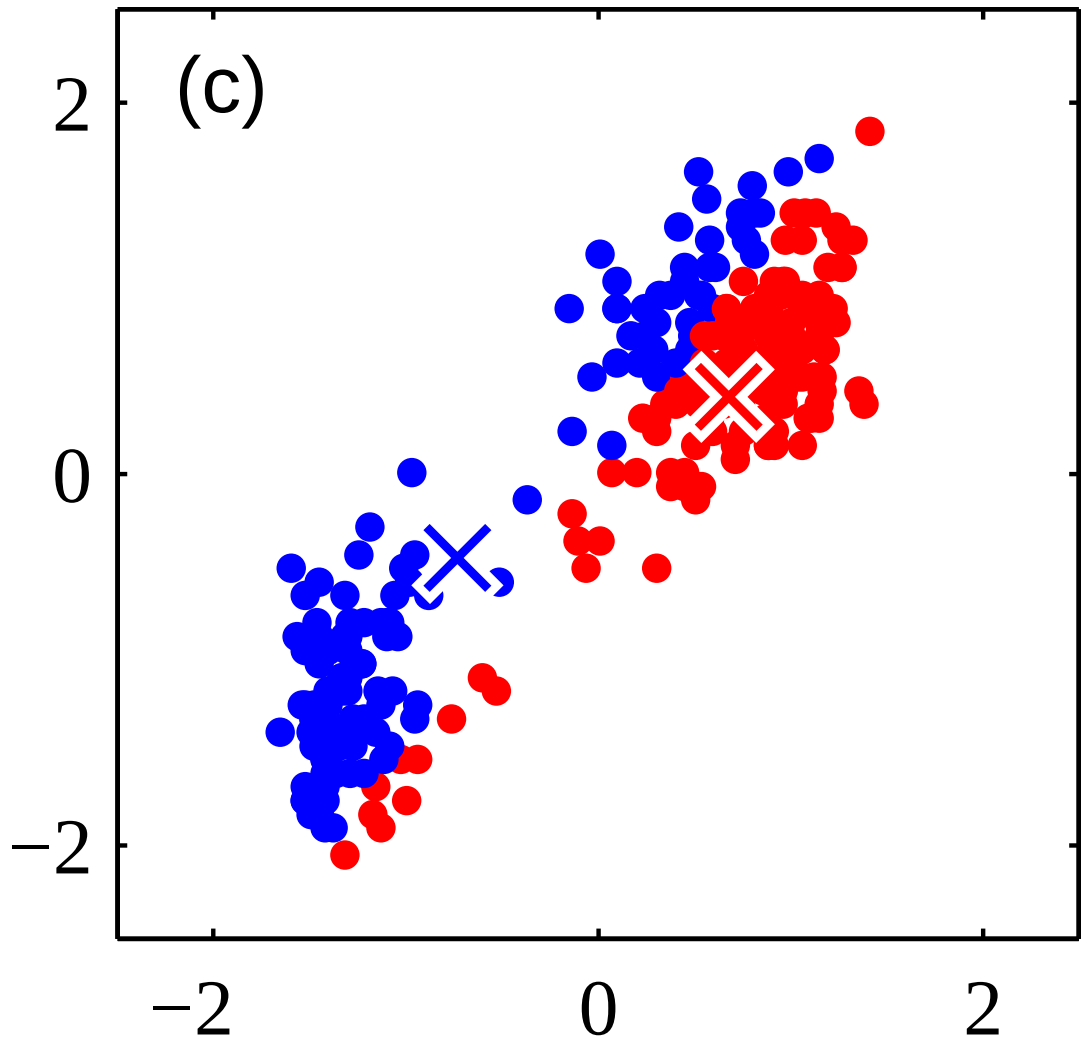
Exemplo



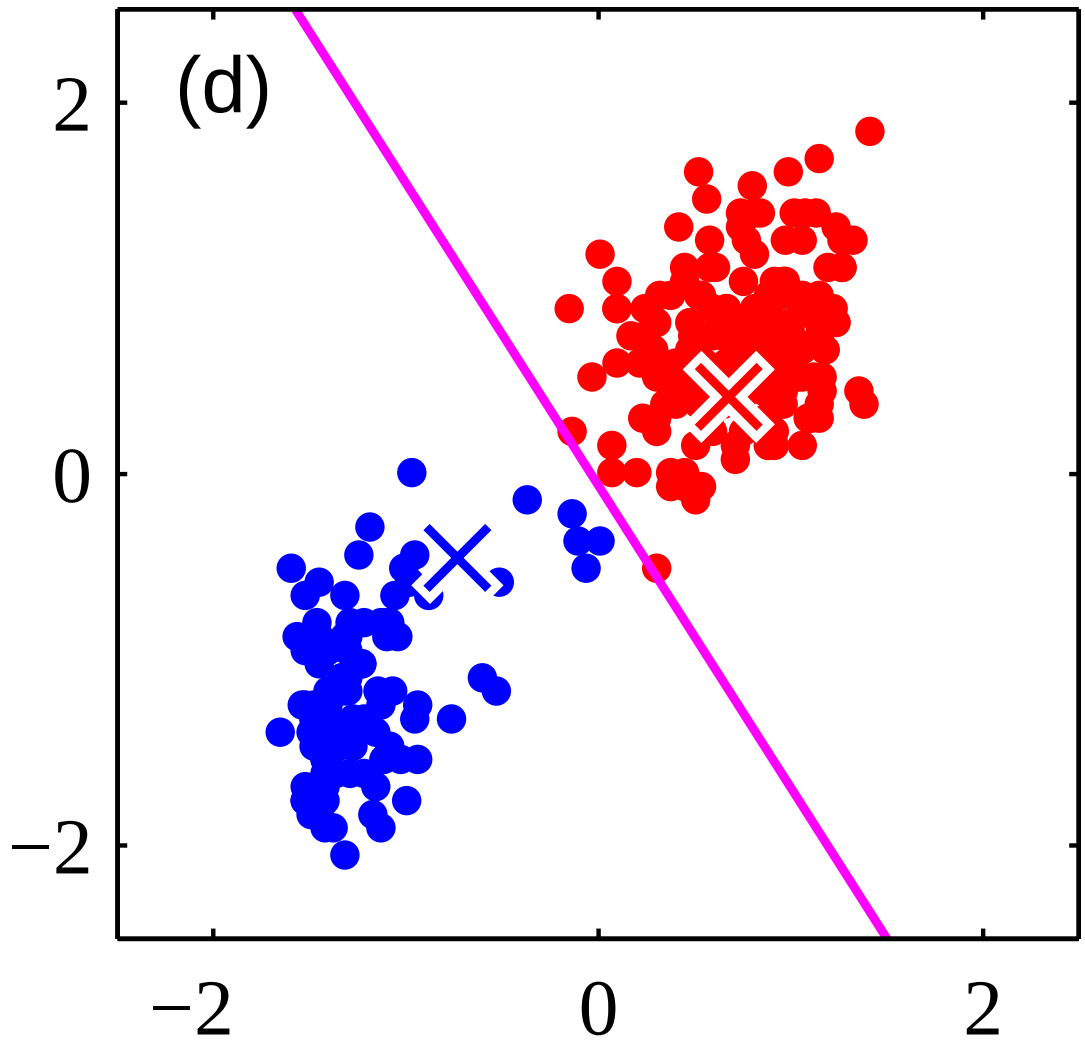
Exemplo



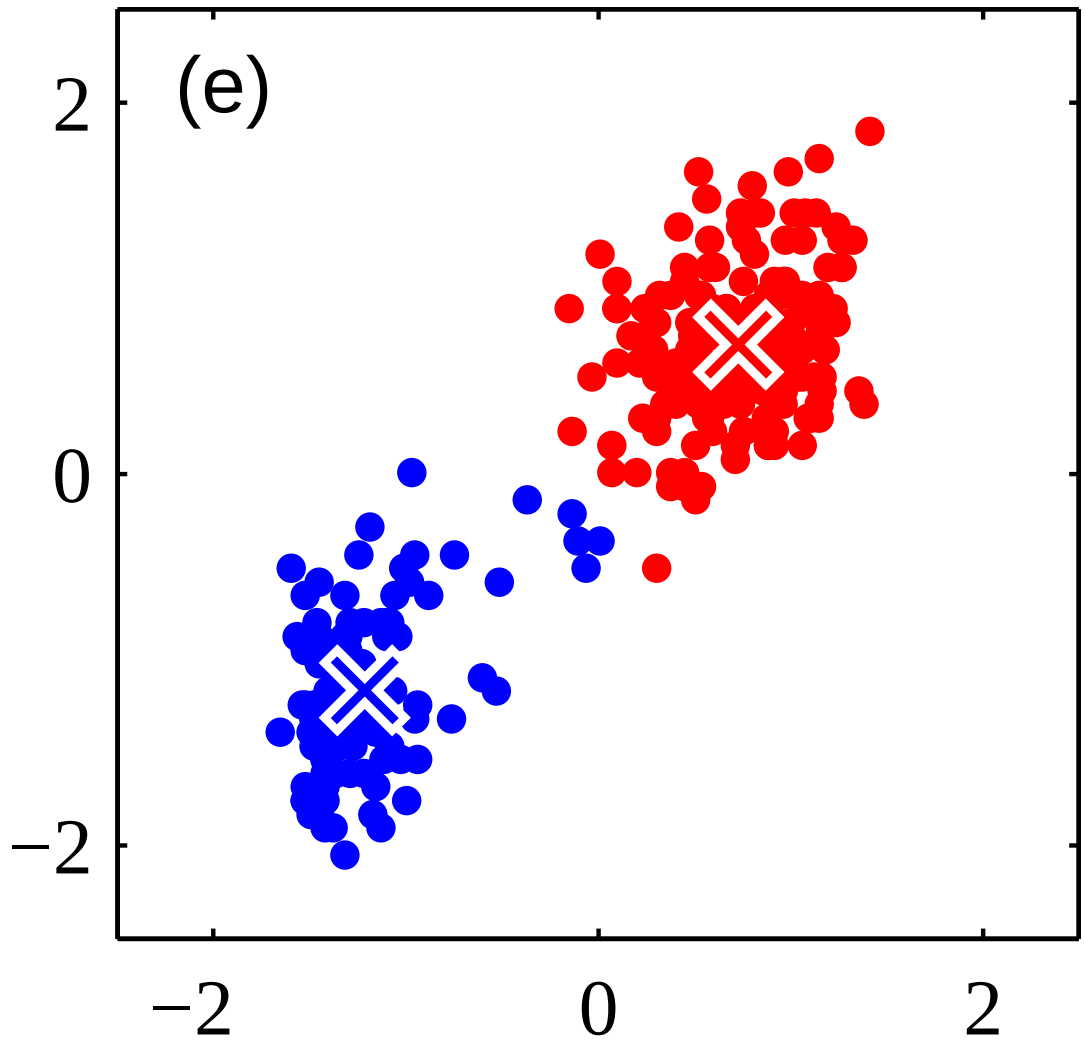
Exemplo



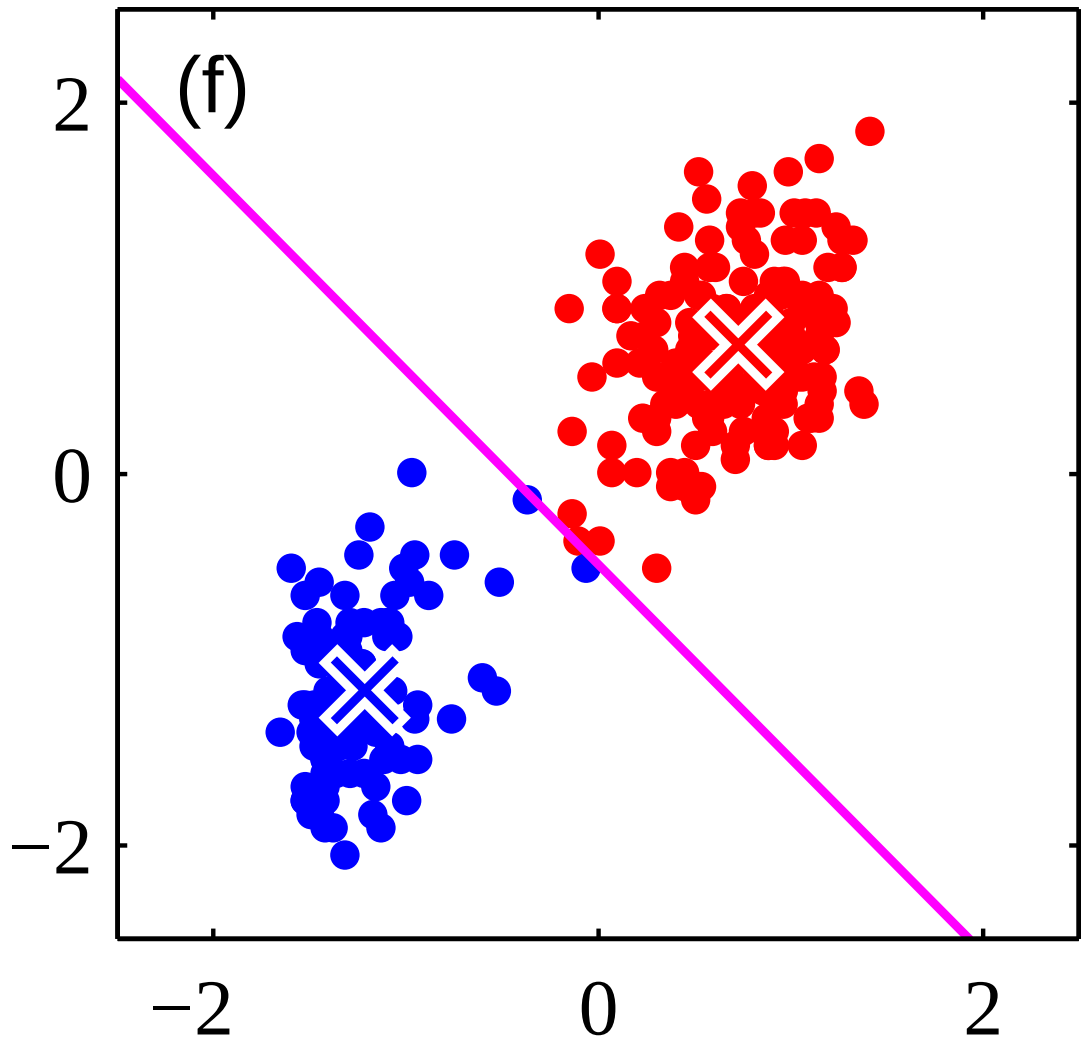
Exemplo



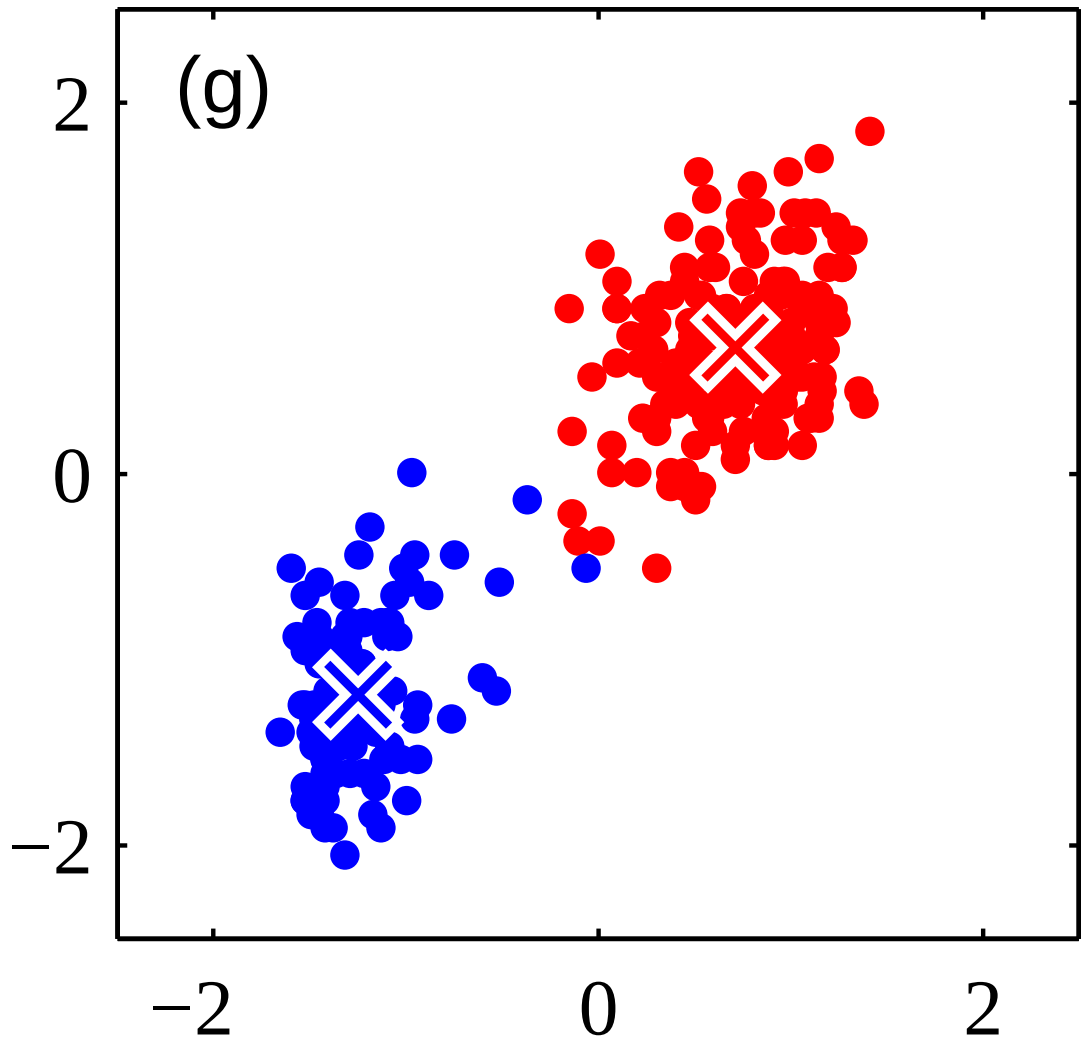
Exemplo



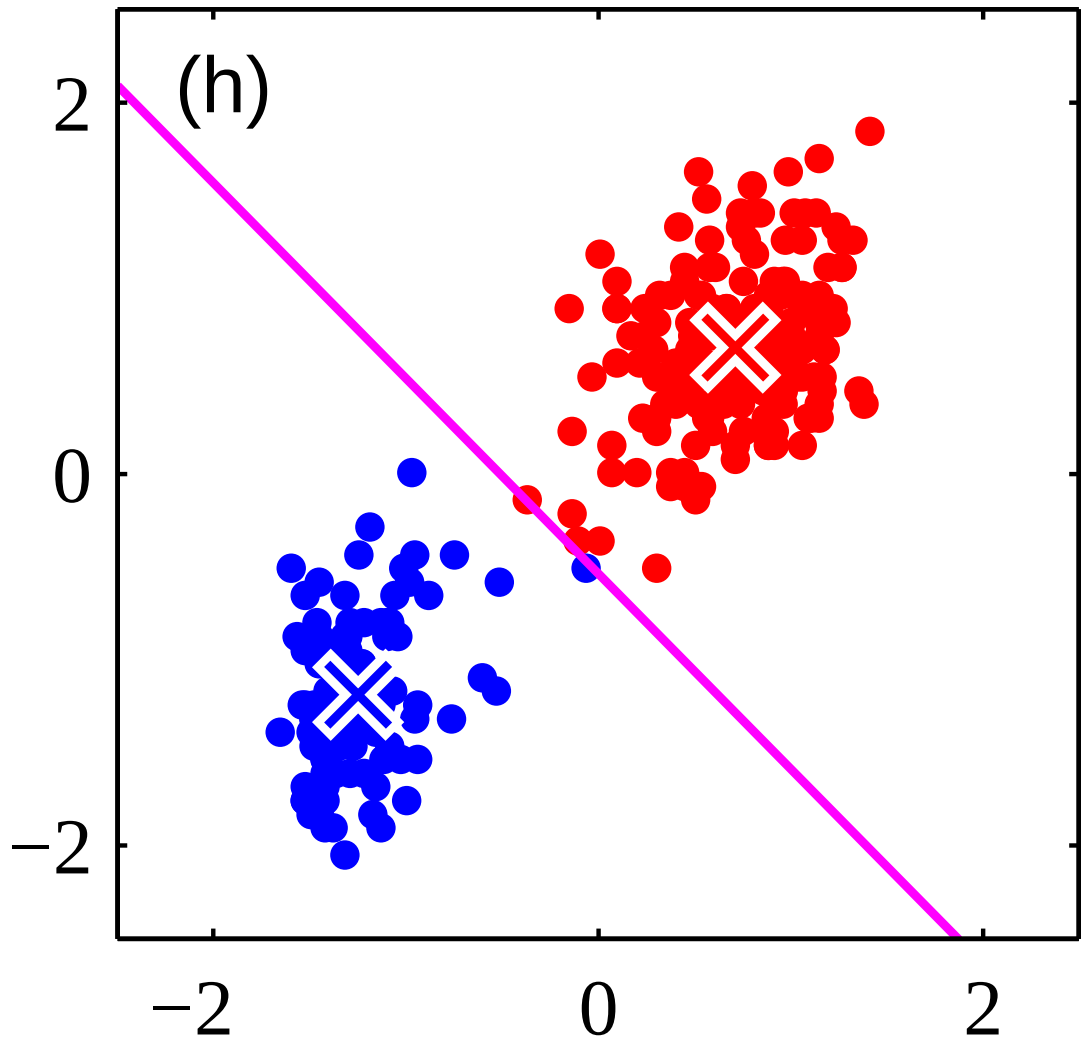
Exemplo



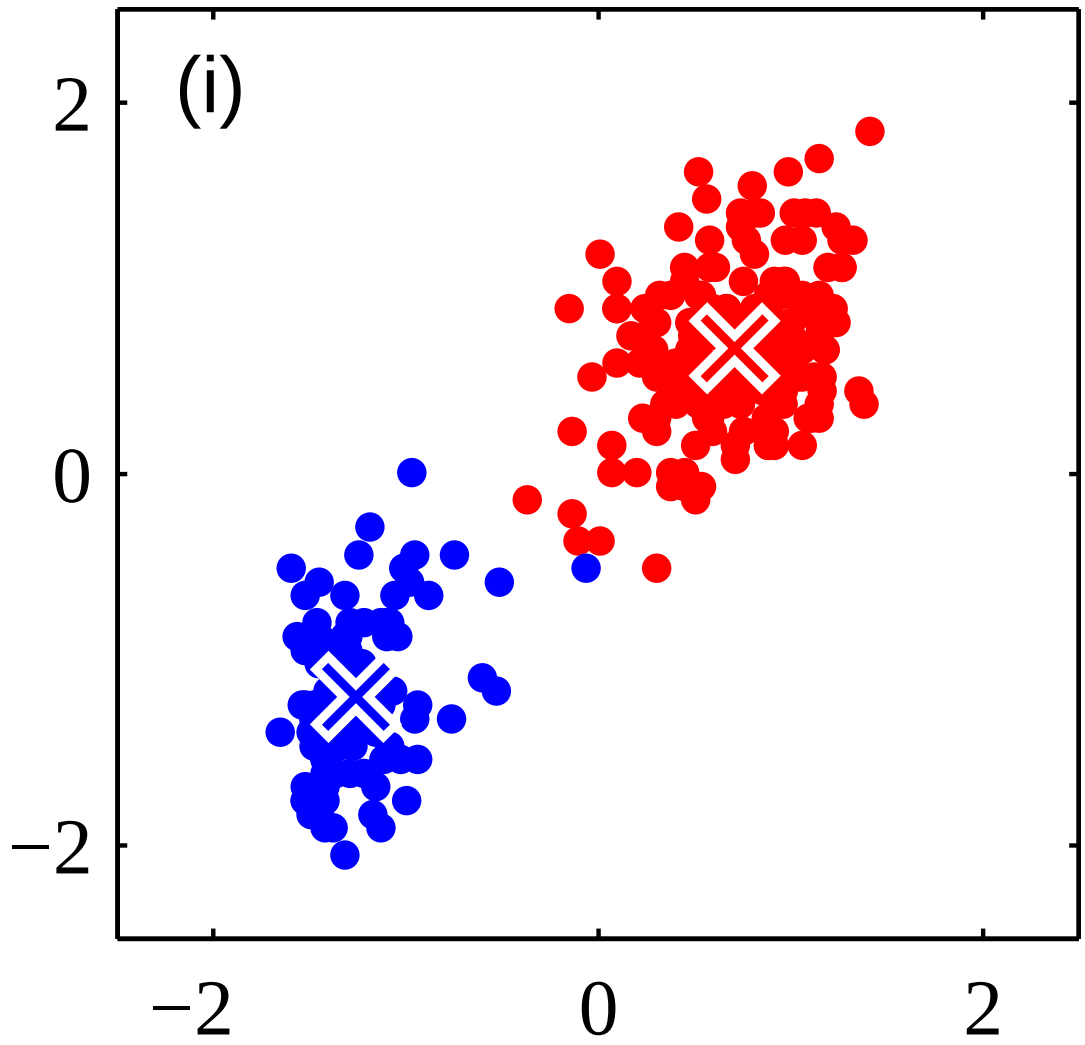
Exemplo



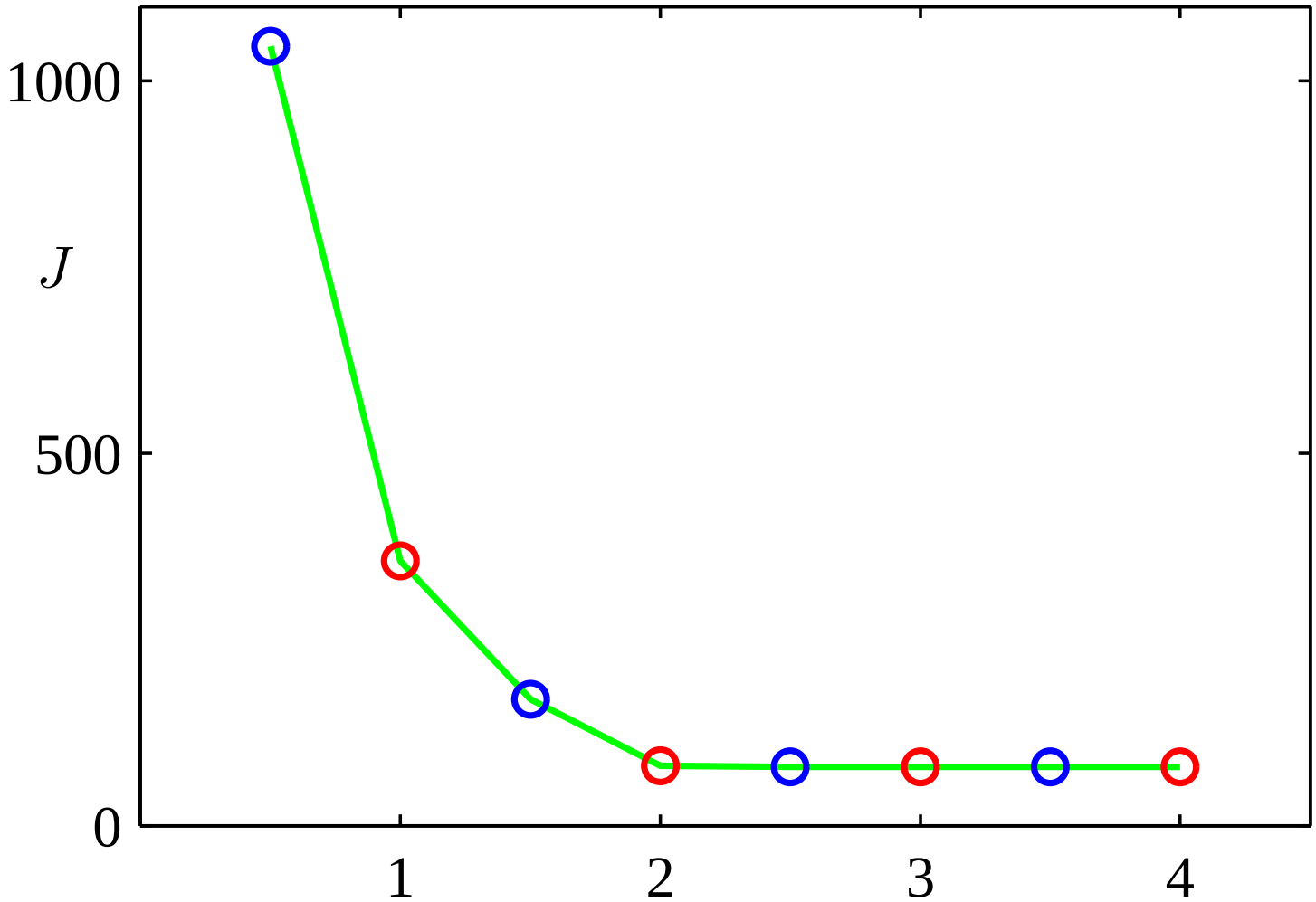
Exemplo



Exemplo



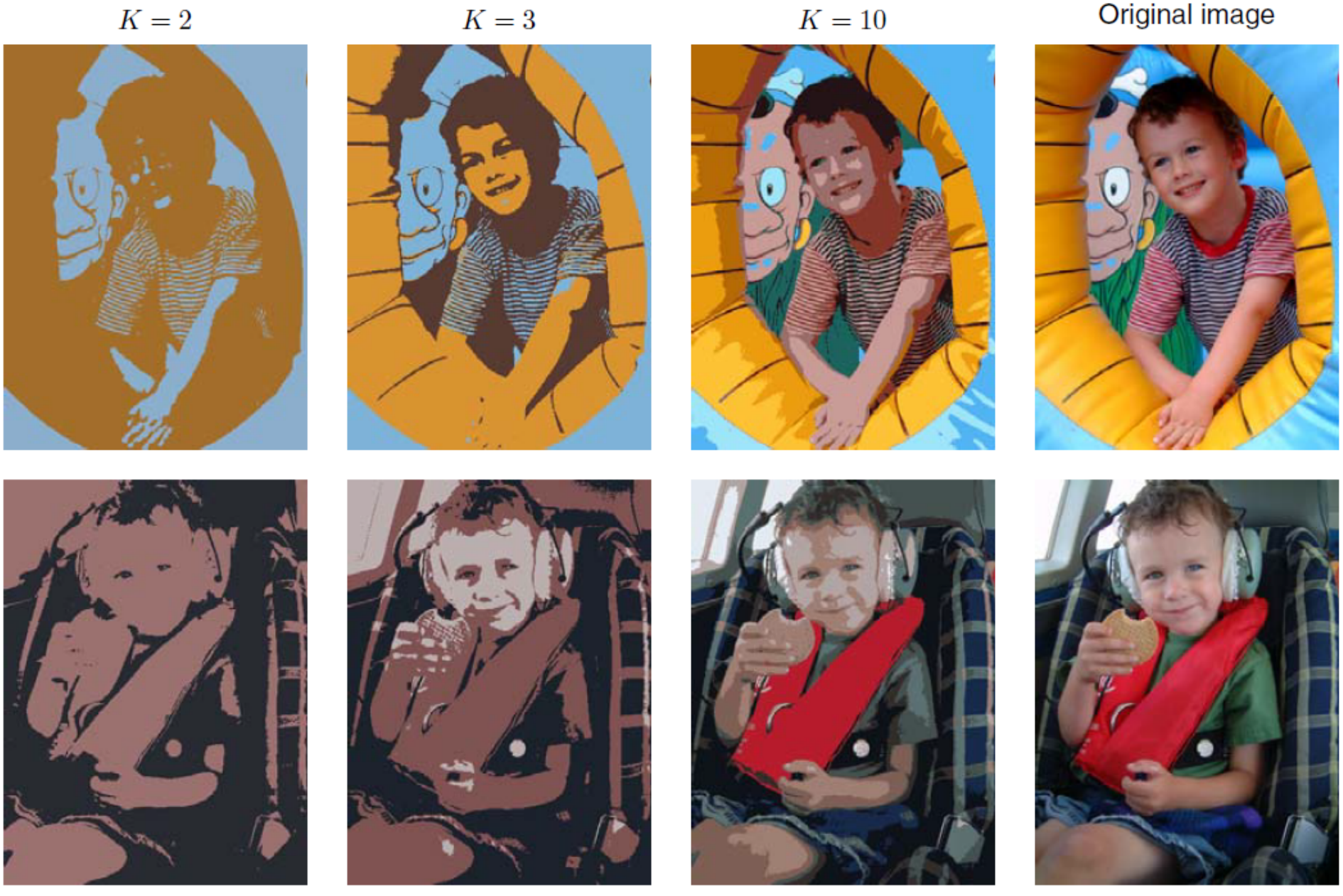
Exemplo: Função Custo



Escolha de K

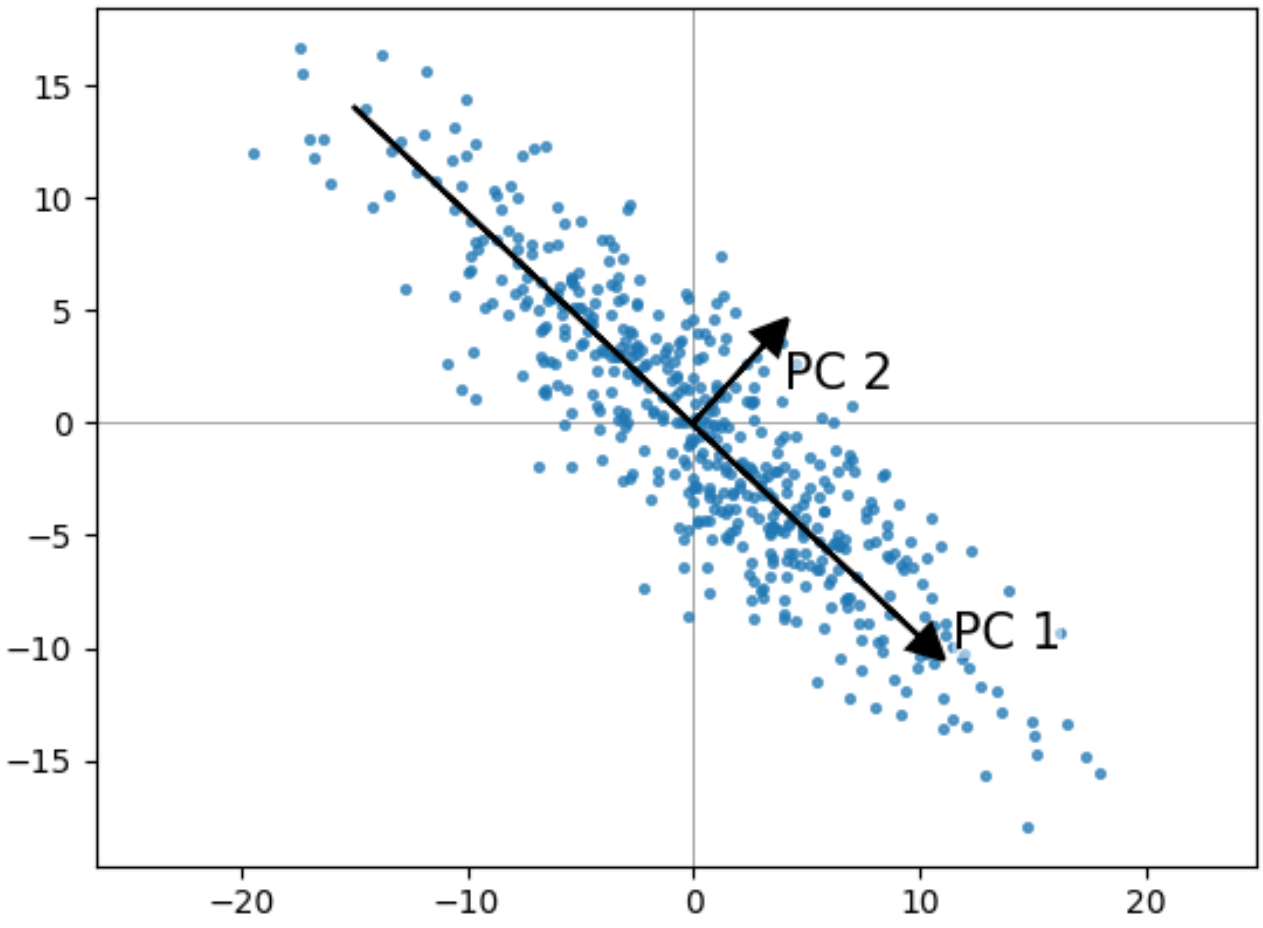
- Considerada uma das principais limitações do algoritmo k-means
- Diversas métricas propostas na literatura
- Se existe uma tarefa final, a métrica pode ser o desempenho na tarefa final

Exemplo: Segmentação/Compressão de Imagens



Análise de Componentes Principais

Motivação: Redução de Dimensionalidade



Análise de Componentes Principais

- PCA - Principal Component Analysis
  - Técnica de aprendizado não-supervisionado
  - Ignora rótulos y, se houver
- Permitir reduzir a dimensionalidade de um vetor de atributos:

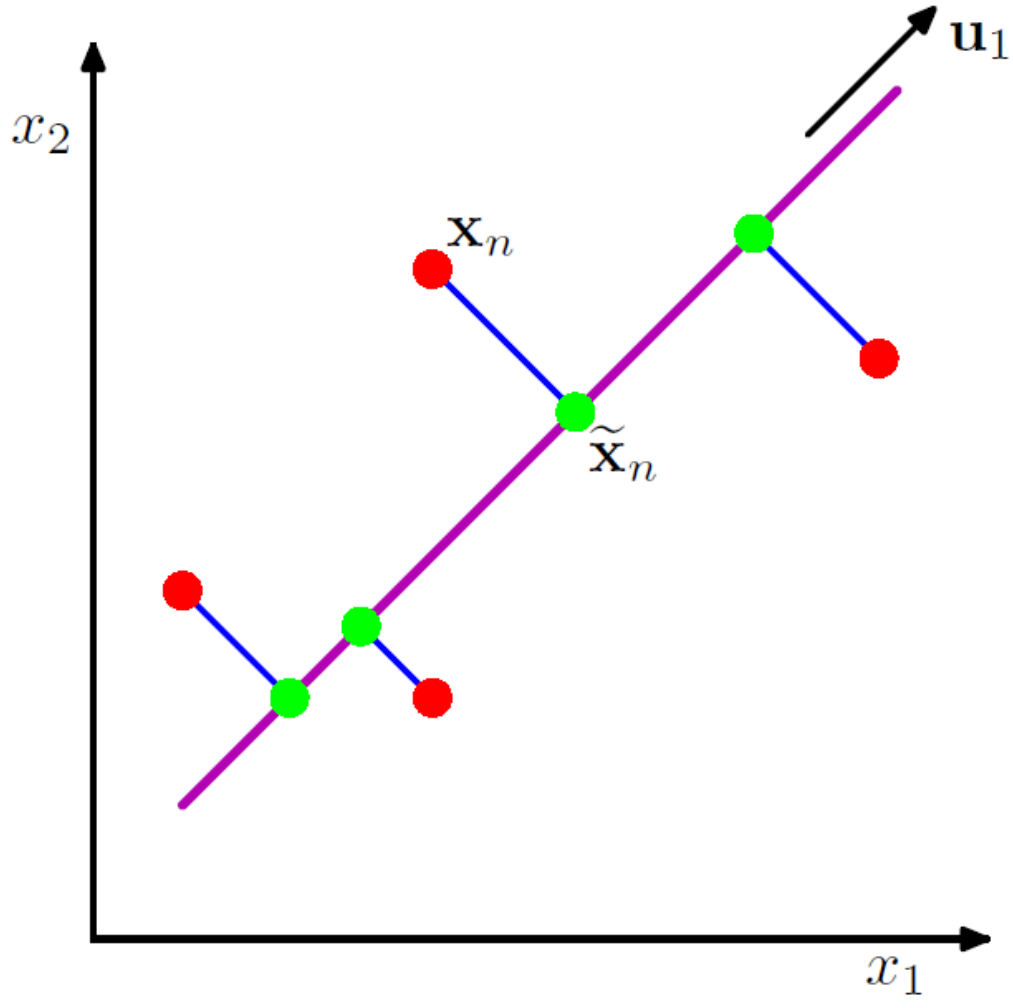
$$\mathbf{x} \in \mathbb{R}^n \rightarrow \mathbf{z} \in \mathbb{R}^k$$

  (idealmente com mínima degradação em relação ao vetor original)
- Principais aplicações:
  - Facilitar a visualização (geralmente em 2-D)
  - Pré-processamento para acelerar algoritmos de aprendizado

Princípios Gerais

- Procedimento básico:
  - Expressar $\mathbf{x}$ em uma nova base ortonormal ($=$ rotacionar/refletir)
  - Reter apenas as $k < n$ primeiras coordenadas do vetor nessa nova base
- Corresponde a escolher um subespaço k-dimensional (gerado pelos k primeiros vetores da nova base) e projetar $\mathbf{x}$ nesse subespaço
- Base deve ser escolhida de forma a minimizar o erro quadrático (médio) entre $\mathbf{x}$ e sua projeção $\hat{\mathbf{x}}$
- Também pode ser interpretado como maximizando a variância da projeção

Interpretação Geométrica



Formulação Matemática

- Suponha que os atributos $x_1, \ldots, x_n$ estão centralizados (média nula) e possuem aproximadamente a mesma variância, i.e.,

$$\mu_{x_j} = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} = 0 \qquad \text{e} \qquad \sigma_{x_j}^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_{x_j})^2 \approx 1$$

- Caso contrário, sempre podemos normalizá-los fazendo

$$x'_j = \frac{x_j - \mu_{x_j}}{\sigma_{x_j}}$$

  - A centralização é útil para simplificar o desenvolvimento matemático, enquanto o escalonamento é importante para obter um bom desempenho na prática

Formulação Matemática

- Seja $\mathbf{u}_1, \ldots, \mathbf{u}_n \in \mathbb{R}^n$ uma base ortonormal, i.e.,

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

- Podemos expressar

$$\mathbf{x}^{(i)} = \sum_{i=1}^{n} c_j^{(i)} \mathbf{u}_j, \qquad \text{onde} \qquad c_j^{(i)} = \mathbf{u}_j^T \mathbf{x}^{(i)}$$

- Matricialmente,

$$\mathbf{x}^{(i)} = \mathbf{U}\mathbf{c}^{(i)}, \qquad \text{onde} \qquad \mathbf{c}^{(i)} = \mathbf{U}^T \mathbf{x}^{(i)},$$

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}, \qquad \mathbf{U} = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_n \\ | & & | \end{bmatrix}, \qquad \mathbf{c}^{(i)} = \begin{bmatrix} c_1^{(i)} \\ \vdots \\ c_n^{(i)} \end{bmatrix}$$

Formulação Matemática

- Projetando no subespaço gerado por $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$, onde $k < n$:

$$\hat{\mathbf{x}}^{(i)} = \sum_{i=1}^{k} c_j^{(i)} \mathbf{u}_j, \qquad \text{onde} \qquad c_j^{(i)} = \mathbf{u}_j^T \mathbf{x}^{(i)}$$

- Matricialmente,

$$\hat{\mathbf{x}}^{(i)} = \hat{\mathbf{U}} \mathbf{z}^{(i)}, \qquad \text{onde} \qquad \mathbf{z}^{(i)} = \hat{\mathbf{U}}^T \mathbf{x}^{(i)},$$

$$\hat{\mathbf{U}}^T \hat{\mathbf{U}} = \mathbf{I}, \qquad \hat{\mathbf{U}} = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_k \\ | & & | \end{bmatrix}, \qquad \mathbf{z}^{(i)} = \begin{bmatrix} z_1^{(i)} \\ \vdots \\ z_k^{(i)} \end{bmatrix} = \begin{bmatrix} c_1^{(i)} \\ \vdots \\ c_k^{(i)} \end{bmatrix}$$

Formulação Matemática

- Desejamos minimizar o erro quadrático médio da projeção:

$$J = \frac{1}{m}\sum_{i=1}^{m} \|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\|^2 = \frac{1}{m}\sum_{i=1}^{m} \|\mathbf{x}^{(i)} - \hat{\mathbf{U}}\mathbf{z}^{(i)}\|^2$$

  onde $\mathbf{z}^{(i)} = \hat{\mathbf{U}}^T\mathbf{x}^{(i)}$, sujeito a restrição $\hat{\mathbf{U}}^T\hat{\mathbf{U}} = \mathbf{I}_k$

- Pode-se mostrar que

$$\|\mathbf{x}^{(i)} - \hat{\mathbf{U}}\mathbf{z}^{(i)}\|^2 = \|\mathbf{x}^{(i)}\|^2 - \|\mathbf{z}^{(i)}\|^2$$

- Portanto, o problema equivale a maximizar a variância da projeção

$$\frac{1}{m}\sum_{i=1}^{m} \|\mathbf{z}^{(i)}\|^2 = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} (z_j^{(i)})^2 = \sum_{j=1}^{k} \sigma_{z_j}^2$$

Formulação Matemática

- Pode-se mostrar que

$$\sum_{j=1}^{k} \sigma_{z_j}^2 = \sum_{j=1}^{k} \mathbf{u}_j^T \mathbf{S} \mathbf{u}_j$$

onde

$$\mathbf{S} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}^{(i)} \mathbf{x}^{(i)T} = \frac{1}{m} \mathbf{X}^T \mathbf{X}$$

é a matriz de covariância (amostral) de $\mathbf{x}$

- A solução ótima da maximização é dada pelos autovetores de $\mathbf{S}$ associados aos k maiores autovalores, obtidos pela decomposição:

$$\mathbf{S} = \mathbf{U} \Lambda \mathbf{U}^T$$

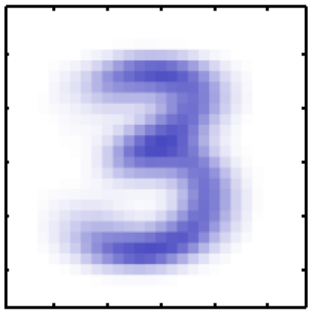
onde $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_1 \geq \cdots \geq \lambda_n$ e $\mathbf{U}^T \mathbf{U} = I$.

  - Extrai-se as k primeiras colunas: $\hat{\mathbf{U}} = \mathbf{U}_{(1:k)} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_k \end{bmatrix}$

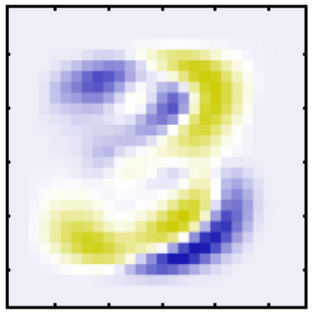
Exemplo

- Autovetores:

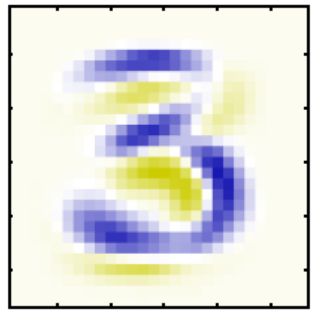
Mean



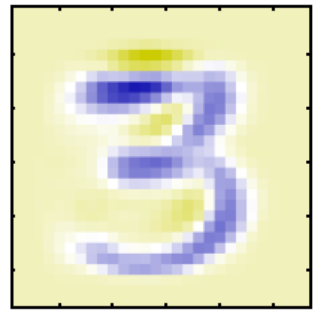
$\lambda_1 = 3.4 \cdot 10^5$



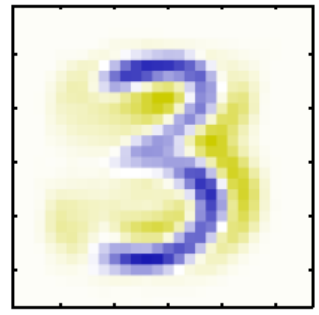
$\lambda_2 = 2.8 \cdot 10^5$



$\lambda_3 = 2.4 \cdot 10^5$

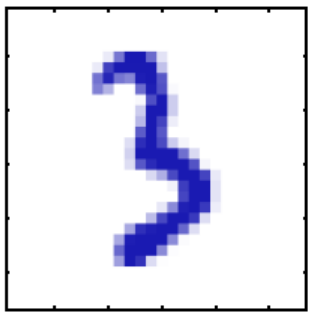


$\lambda_4 = 1.6 \cdot 10^5$

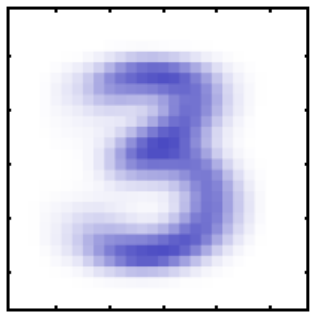


- Reconstrução ($M = k$):

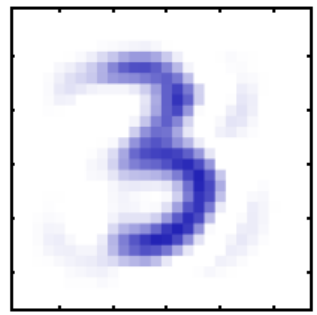
Original



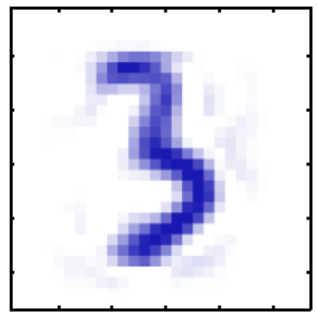
$M = 1$



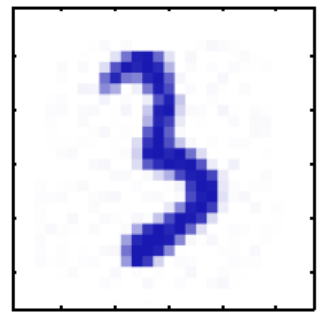
$M = 10$



$M = 50$



$M = 250$



Decomposição em Valores Singulares

- A decomposição em valores singulares (SVD) de $\mathbf{X}^T$ é dada por:

$$\mathbf{X}^T = \mathbf{U}\Sigma\mathbf{V}^T$$

  - $\mathbf{U} \in \mathbb{R}^{n\times n}$ satisfaz $\mathbf{U}^T\mathbf{U} = \mathbf{I}$
  - $\Sigma \in \mathbb{R}^{n\times m}$ é diagonal com elementos não-negativos
  - $\mathbf{V} \in \mathbb{R}^{m\times m}$ satisfaz $\mathbf{V}^T\mathbf{V} = \mathbf{I}$

- Consequentemente,

$$\mathbf{S} = \frac{1}{m}\mathbf{X}^T\mathbf{X} = \frac{1}{m}\mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma^T\mathbf{U}^T = \frac{1}{m}\mathbf{U}\Sigma\Sigma^T\mathbf{U}^T$$

- Logo, a decomposição em autovalores e autovetores de $\mathbf{S}$ é

$$\mathbf{S} = \frac{1}{m}\mathbf{X}^T\mathbf{X} = \mathbf{U}\Lambda\mathbf{U}^T, \qquad \text{onde } \Lambda = \frac{1}{m}\Sigma\Sigma^T$$

  que por definição também é igual a SVD de $\mathbf{S}$.

Decomposição em Valores Singulares

- Caso $n > m$, teremos $\lambda_j = 0$ para todo $j > m$, i.e.,

$$\Lambda_{(n\times n)} = \begin{bmatrix} \tilde{\Lambda}_{(m\times m)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m, 0, \ldots, 0)$$

- Nesse caso,

$$\frac{1}{m}\mathbf{X}\mathbf{X}^T = \frac{1}{m}\mathbf{V}\Sigma^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T = \frac{1}{m}\mathbf{V}\Sigma^T\Sigma\mathbf{V}^T = \mathbf{V}\tilde{\Lambda}\mathbf{V}^T$$

  e portanto,

$$\mathbf{S}(\mathbf{X}^T\mathbf{V}) = \frac{1}{m}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{V} = \mathbf{X}^T\mathbf{V}\tilde{\Lambda}\mathbf{V}^T\mathbf{V} = (\mathbf{X}^T\mathbf{V})\tilde{\Lambda}$$

- Podemos encontrar $\mathbf{U}_{(1:m)}$ normalizando os autovetores $\mathbf{X}^T\mathbf{V}$:

$$\mathbf{U}_{(1:m)} = \frac{1}{\sqrt{m}}\mathbf{X}^T\mathbf{V}\tilde{\Lambda}^{-\frac{1}{2}}$$

  de forma que $\mathbf{U}_{(1:m)}^T\mathbf{U}_{(1:m)} = \mathbf{I}$ (as demais colunas são irrelevantes)

Cálculo da Matriz de Transformação

- SVD de $\mathbf{X}^T$:

$$(\mathbf{U},\ \Sigma,\ \_) = \texttt{svd}(\mathbf{X}^T,\ \texttt{full\_matrices = False})$$

$$(\lambda_1,\ \ldots,\ \lambda_n) = \texttt{diag}(\Sigma\Sigma^T/m)$$

  - `full_matrices = False`: opcional, reduz a complexidade se $n \neq m$

- Se $m \gg n$, é mais eficiente calcular a SVD de $\mathbf{S} \in \mathbb{R}^{n \times n}$:

$$(\mathbf{U},\ \Lambda,\ \_) = \texttt{svd}(\mathbf{X}^T\mathbf{X}/m)$$

$$(\lambda_1,\ \ldots,\ \lambda_n) = \texttt{diag}(\Lambda)$$

- Se $n \gg m$, é mais eficiente calcular a SVD de $\mathbf{X}\mathbf{X}^T/m \in \mathbb{R}^{m \times m}$:

$$(\mathbf{V},\ \tilde{\Lambda},\ \_) = \texttt{svd}(\mathbf{X}\mathbf{X}^T/m)$$

$$\mathbf{U}_{(1:m)} = \mathbf{X}^T\mathbf{V}(m\tilde{\Lambda})^{-\frac{1}{2}}$$

$$(\lambda_1,\ \ldots,\ \lambda_m) = \texttt{diag}(\tilde{\Lambda})$$

Escolha do número de componentes principais

- Total de variância nos dados originais:

$$E_{\mathbf{x}} = \sum_{j=1}^{n} \sigma_{x_j}^2 = \mathrm{tr}\left(\frac{1}{m}\mathbf{X}^T\mathbf{X}\right) = \sum_{j=1}^{n} \lambda_j$$

- Total de variância "retida" (pelos dados em dimensão reduzida):

$$E_{\mathbf{z}} = \sum_{j=1}^{k} \sigma_{z_j}^2 = \mathrm{tr}\left(\frac{1}{m}\mathbf{Z}^T\mathbf{Z}\right) = \sum_{j=1}^{k} \lambda_j$$

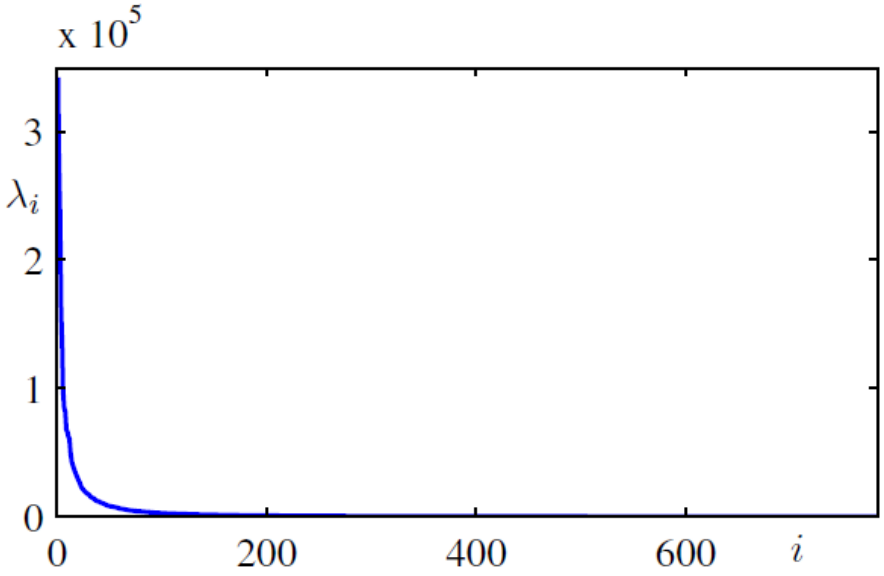
- Percentual de "variância retida":

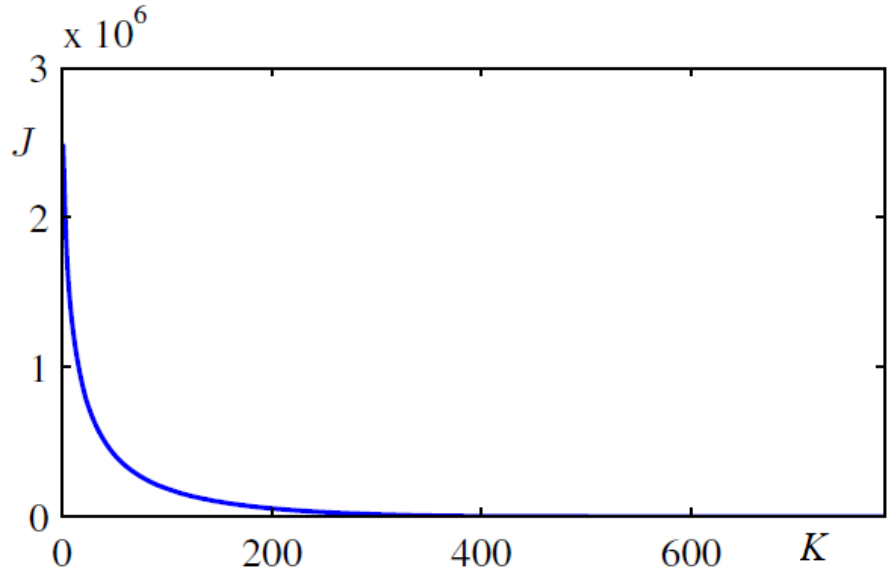
$$\frac{E_{\mathbf{z}}}{E_{\mathbf{x}}} = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{n} \lambda_j}$$

- Escolhe-se k tal que $E_{\mathbf{z}}/E_{\mathbf{x}}$ seja maior que um dado valor (ex: 90%)

Escolha do número de componentes principais

Exemplo:





- Também pode-se analisar o custo (erro quadrático médio da aproximação):

$$J = \frac{1}{m} \sum_{i=1}^{m} \|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\|^2 = \sum_{j=k+1}^{n} \lambda_j$$

Aplicação: Aceleração de Algoritmos de Aprendizado

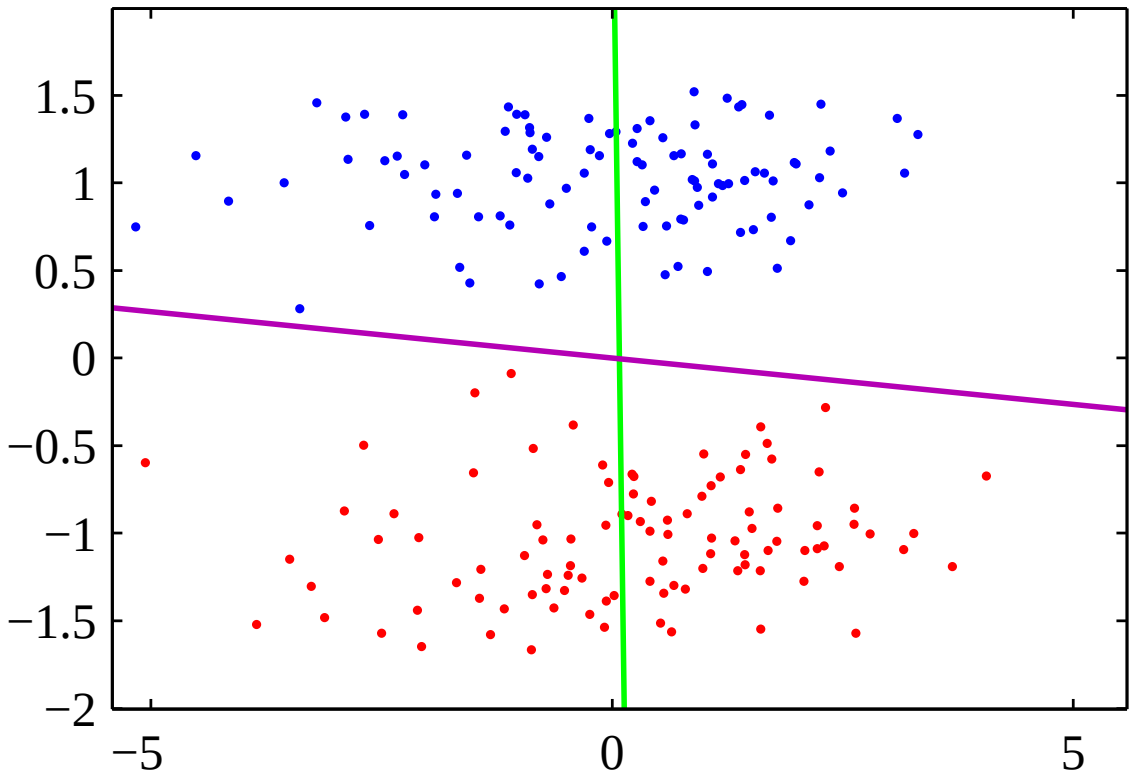
- Se um conjunto de dados possui um número muito grande de atributos, tornando o aprendizado excessivamente lento, pode-se considerar a aplicação de PCA para reduzir a dimensionalidade

- Exemplo: Aprendizado Supervisionado
  - Conjunto de treinamento: $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})$
  - $\mathbf{x}^{(i)} \in \mathbb{R}^n \quad \stackrel{\text{PCA}}{\longrightarrow} \quad \mathbf{z}^{(i)} = \hat{\mathbf{U}}^T \mathbf{x}^{(i)} \in \mathbb{R}^k$
  - Novo conjunto de treinamento: $(\mathbf{z}^{(1)}, y^{(1)}), \ldots, (\mathbf{z}^{(m)}, y^{(m)})$

- O mapeamento $\mathbf{x} \rightarrow \mathbf{z}$ (o que envolve o cálculo de $\mu_{\mathbf{x}}$, $\sigma_{\mathbf{x}}$ e $\hat{\mathbf{U}}$) deve ser definido a partir do conjunto de treinamento e aplicado (sem alteração) no conjunto de teste

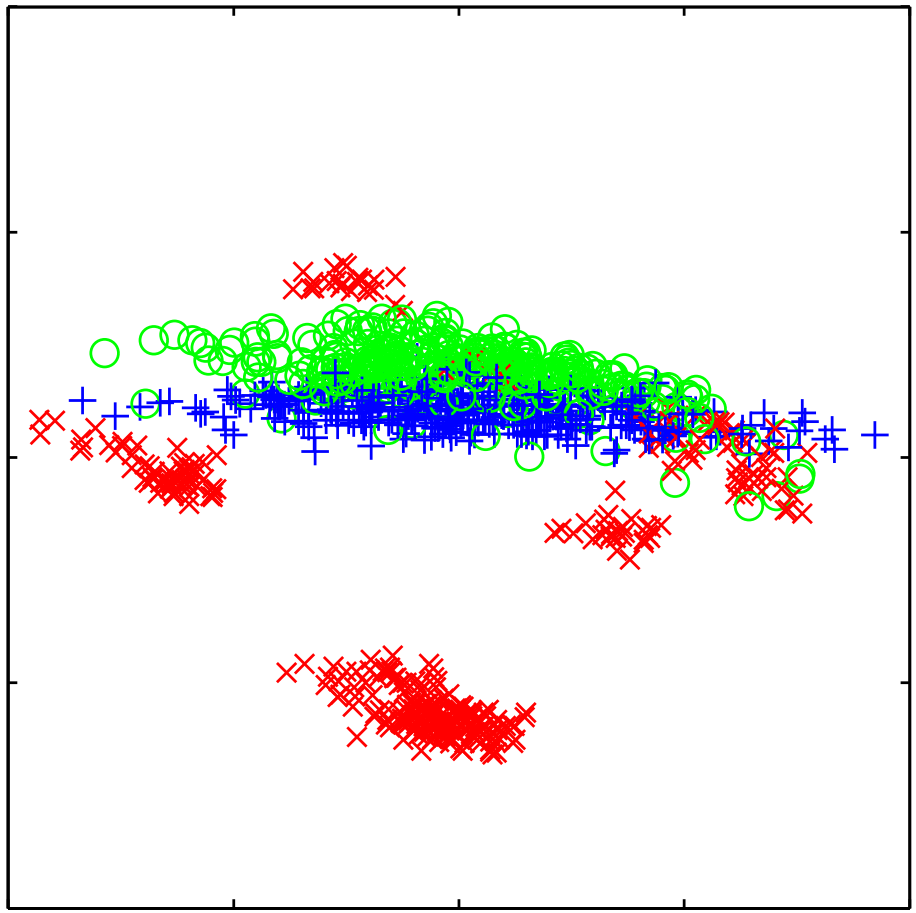
- PCA é um método não-supervisionado: ignora rótulos $y^{(i)}$
  - Versões supervisionadas existem (*Supervised PCA*) que tentam levar em conta também o desempenho na tarefa

Exemplo



- A direção de máxima variância não necessariamente fornece a melhor separação entre classes

Exemplo: Visualização



- Dimensão original $n = 12 \Longrightarrow k = 2$

Exemplo: *Eigenfaces*

Amostras:

Exemplo: *Eigenfaces*

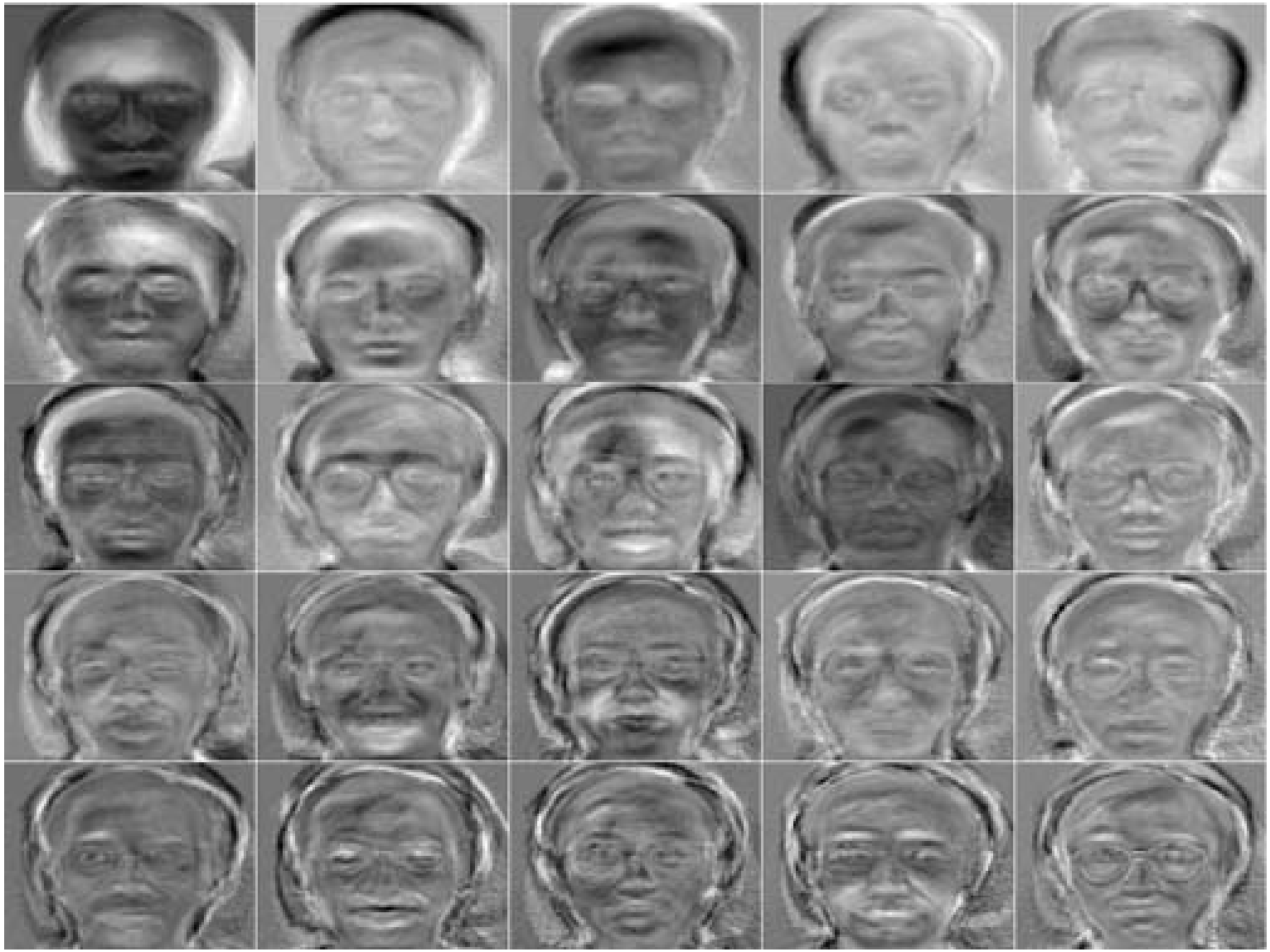
Média e componentes principais:

Exemplo: *Eigenfaces*

Original e reconstrução:

Exemplo: *Eigenfaces*

Componentes principais:



Exemplo: *Eigenfaces*

Média e reconstrução: