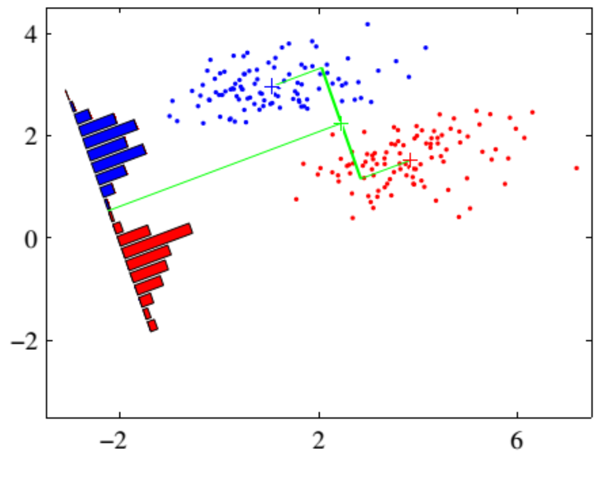
Extração e Seleção de Características Redução de Dimensionalidade

André E. Lazzaretti
UTFPR/CPGEI



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

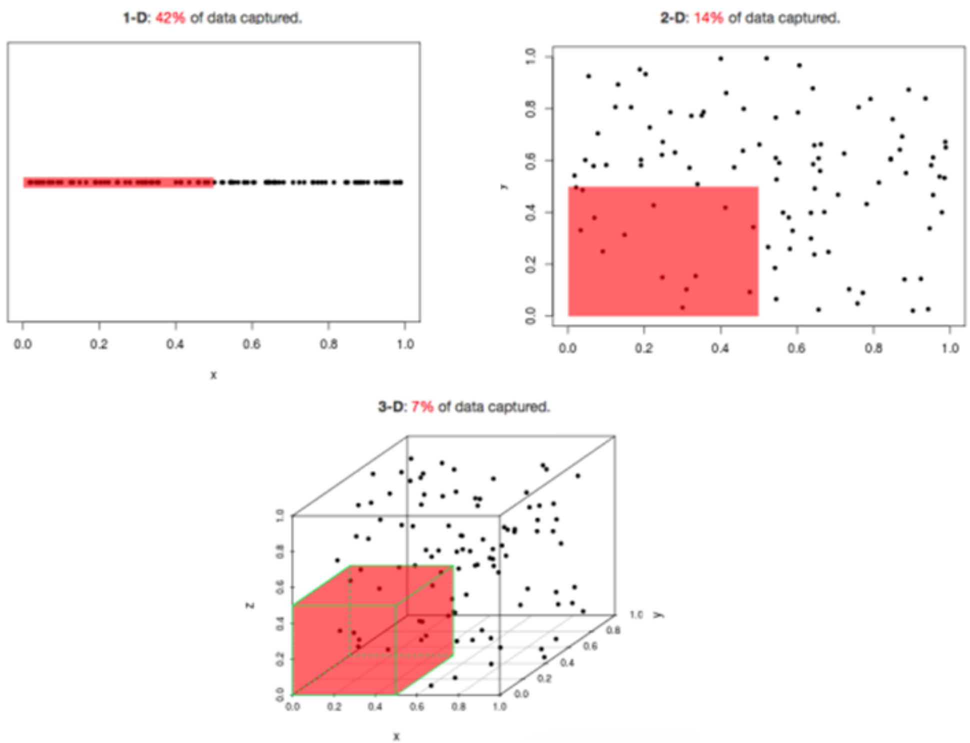
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E INFORMÁTICA INDUSTRIAL

Quais *Features* Utilizar?

- **Altamente dependente do problema:**
  - Imagens:
    - *Histogram of Oriented Gradients* (HOG)
    - *Scale-Invariant Feature Transform* (SIFT)
    - *Bag-of-words*
  - Vídeos:
    - Fluxo óptico (histogramas relacionados)
    - *Tracking* (Filtro de Kalman, Filtro de Partículas)
  - Audio/Speech Recognition:
    - FFT, Mel *Frequency Cepstral Coefficient*
    - Transformada *Wavelet*
- ... Lista enorme -> foco no seu problema!
- Aprender as features:
  - Abordagens de Deep Learning e Redes Neurais Convolucionais
  - k-SVD

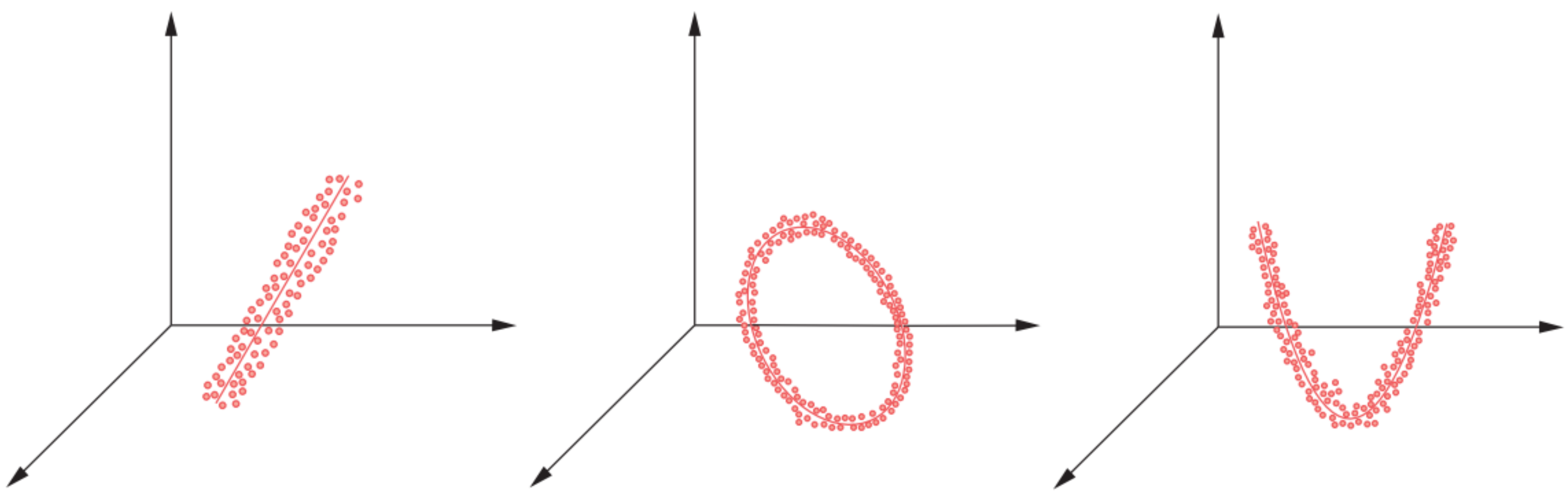
Redução de Dimensionalidade

- *Curse of dimensionality*: O número de exemplos que são capturados por algum comprimento fixo diminui rapidamente à medida que a dimensão aumenta.



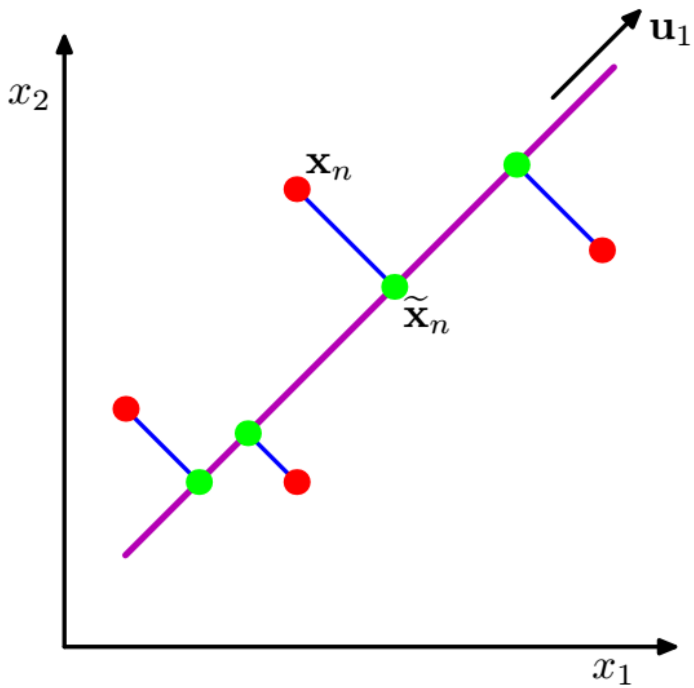
Redução de Dimensionalidade e Feature Extraction

- Em muitas aplicações práticas, embora os dados estejam representados em um espaço de alta dimensionalidade, a verdadeira dimensionalidade, conhecida como **dimensionalidade intrínseca**, pode ser de um valor muito menor:



Principal Component Analysis

- **Ideia**: projetar (ortogonalmente) os dados em um novo espaço (linear) que maximize a variância dos dados projetados.



Principal Component Analysis

- Dado um conjunto de dados $\{\mathbf{x}_n\}_1^N$ sendo d a dimensionalidade de $\mathbf{x}_n$, o objetivo é projetar os dados em um novo espaço com dimensionalidade $m<d$, maximizando a variância dos dados projetados.
- Considerando inicialmente que $m=1$. O vetor $\mathbf{u}_1$ define a direção neste novo espaço.
- A média dos dados projetados será dada por $\mathbf{u}_1^T\overline{\mathbf{x}}$, sendo $\overline{\mathbf{x}}$ a média dos dados, dada por:

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

Principal Component Analysis

- A variância dos dados projetados é dada por:

$$\frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{u}_1^T\mathbf{x}_n - \mathbf{u}_1^T\overline{\mathbf{x}}\right)^2 = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$$

- Sendo S a matriz de covariância, definida por:

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^T$$

- Reformulando: o objetivo é maximizar a variância dos dados projetados $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$, em relação à $\mathbf{u}_1$.
- Para prevenir que $\|\mathbf{u}_1\|$ tenda a infinito, utiliza-se a restrição: $\mathbf{u}_1^T\mathbf{u}_1 = 1$. Isso resulta na seguinte formulação:

$$\begin{array}{ll} \text{maximize} & \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 \\ \text{subject to} & \mathbf{u}_1^T\mathbf{u}_1 = 1 \end{array}$$

Principal Component Analysis

- Escrevendo o Lagrangeano:

$$L(\mathbf{u}_1, \lambda_1) = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 - \lambda_1(\mathbf{u}_1^T \mathbf{u}_1 - 1)$$

- Derivando e igualando a zero:

$$\frac{\partial L(\mathbf{u}_1, \lambda_1)}{\partial \mathbf{u}_1} = \mathbf{0} \quad \Rightarrow \quad \mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

Autovalor de **A**

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

Autovetor de **A**

- Isso implica que $\mathbf{u}_1$ deve ser um autovetor de **S**.

- A variância é maximizada quando $\mathbf{u}_1$ é igual ao autovetor com o maior autovalor associado λ_1.

Principal Component Analysis

- O segundo autovetor $\mathbf{u}_2$ deve ser ortogonal a $\mathbf{u}_2$:

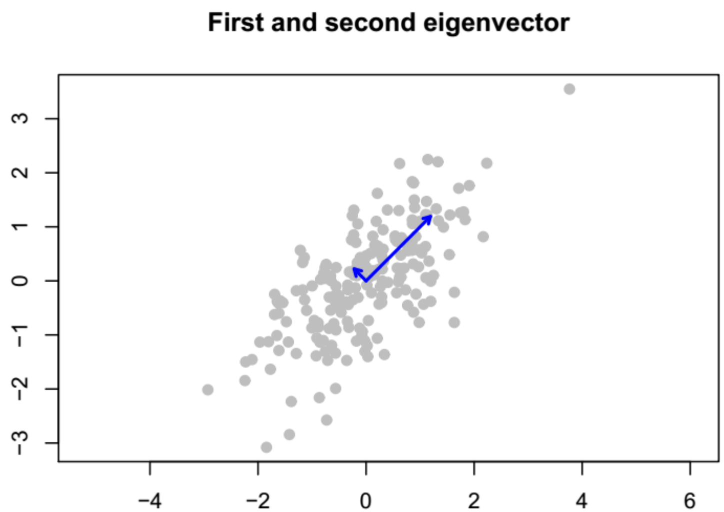
$$\begin{array}{ll} \text{maximize} & \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2 \\ \text{subject to} & \mathbf{u}_2^T \mathbf{u}_2 = 1, \quad \mathbf{u}_2^T \mathbf{u}_1 = 0 \end{array}$$

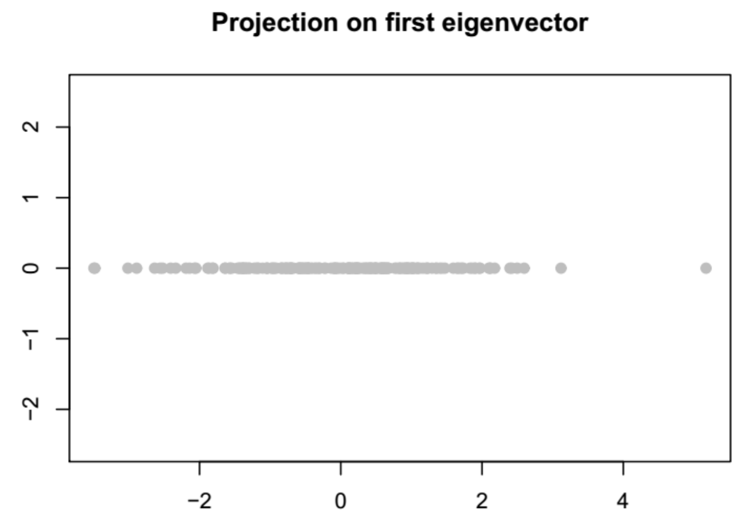
- Organizando o Lagrangeano:

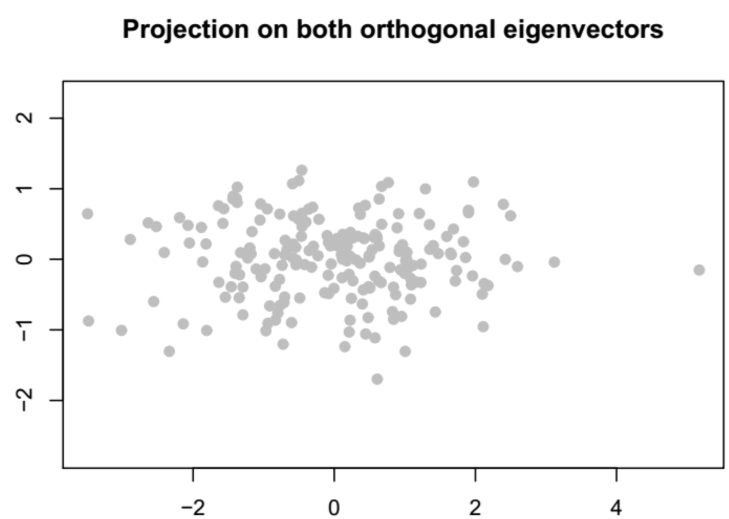
$$L(\mathbf{u}_2, \lambda_1, \lambda_2) = \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2 - \lambda_2(\mathbf{u}_2^T \mathbf{u}_2 - 1) - \lambda_1(\mathbf{u}_2^T \mathbf{u}_1 - 0)$$

- Resulta em: $\mathbf{S}\mathbf{u}_2 = \lambda_2 \mathbf{u}_2$
- O que implica que $\mathbf{u}_2$ deve ser um autovetor de **S** com o segundo maior autovalor λ_2. Outras dimensões de projeção são dadas pelos autovetores com os autovalores decrescentes.
- **Em resumo**: PCA é a decomposição em autovetores e autovalores da matriz de covariância **S=XX**^{T}.

Principal Component Analysis



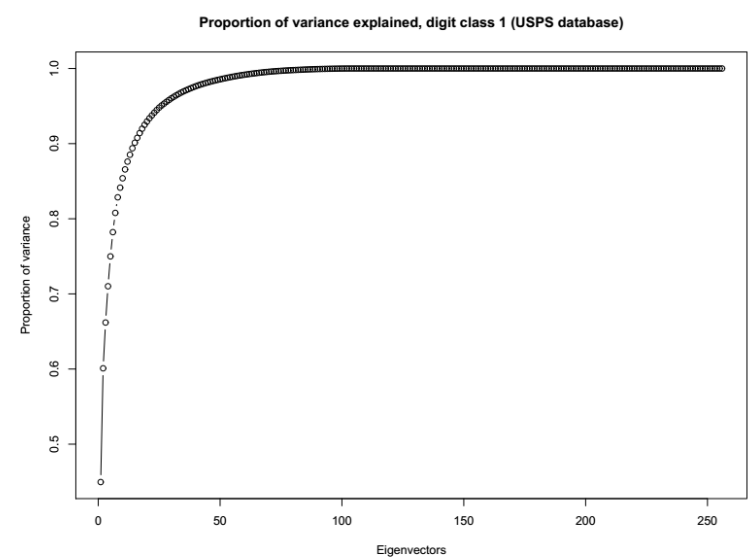




Principal Component Analysis

- No caso de imagens e sinais de voz, as entradas são altamente correlacionadas.
- Se as dimensões estão altamente correlacionadas, então haverá um pequeno número de autovetores com os maiores autovalores (m<<d). Com isso, pode-se obter uma redução relevante na dimensionalidade:

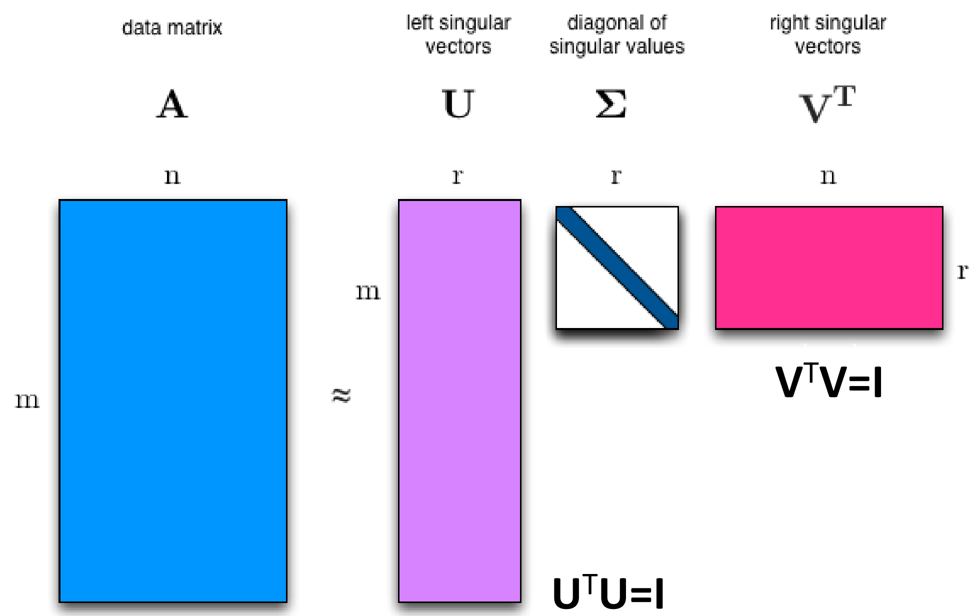
$$\frac{\lambda_1 + \lambda_2 + \ldots + \lambda_m}{\lambda_1 + \lambda_2 + \ldots + \lambda_m + \ldots + \lambda_d}$$



Exemplo MATLAB!

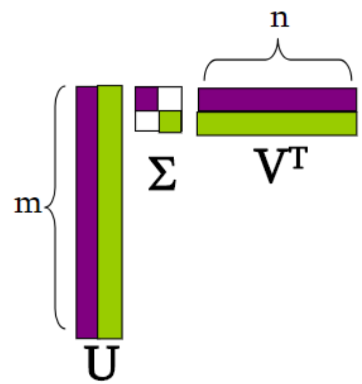
Relação com *Singular Value Decomposition* (SVD)

- *Eigendecomposition* da matriz **S** pode não ser eficiente em alguns casos, ou mesmo inviável;
- Alternativa: SVD!
  - $\mathbf{A} = \mathbf{U\Sigma V}^T$



SVD – User to Movies Example

$$
\begin{array}{c|c}
 & \text{Matrix} \quad \text{Alien} \quad \text{Serenity} \quad \text{Casablanca} \quad \text{Amelie} \\
\begin{array}{c} \text{SciFi} \\ \\ \\ \\ \text{Romance} \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
$$



Seria possível reduzir dimensionalidade só usando SVD?

Sim -> projeção $(U\Sigma)^T$

E a reconstrução?

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\approx
\begin{bmatrix}
0.92 & 0.95 & 0.92 & 0.01 & 0.01 \\
2.91 & 3.01 & 2.91 & -0.01 & -0.01 \\
3.90 & 4.04 & 3.90 & 0.01 & 0.01 \\
4.82 & 5.00 & 4.82 & 0.03 & 0.03 \\
0.70 & 0.53 & 0.70 & 4.11 & 4.11 \\
-0.69 & 1.34 & -0.69 & 4.78 & 4.78 \\
0.32 & 0.23 & 0.32 & 2.01 & 2.01
\end{bmatrix}
$$

scifi

romance

Peso scifi concept

$$
\begin{array}{c|c}
 & \text{Matrix} \quad \text{Alien} \quad \text{Serenity} \quad \text{Casablanca} \quad \text{Amelie} \\
\begin{array}{c} \text{SciFi} \\ \\ \\ \\ \text{Romnce} \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.13 & 0.02 & \cancel{-0.01} \\
0.41 & 0.07 & \cancel{-0.03} \\
0.55 & 0.09 & \cancel{-0.04} \\
0.68 & 0.11 & \cancel{-0.05} \\
0.15 & -0.59 & \cancel{0.65} \\
0.07 & -0.73 & \cancel{-0.67} \\
0.07 & -0.29 & \cancel{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & \cancel{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\
\cancel{0.40} & \cancel{-0.80} & \cancel{0.40} & \cancel{0.09} & \cancel{0.09}
\end{bmatrix}
$$

U is the user-to-concept similarity matrix

V is the movie-to-concept similarity matrix

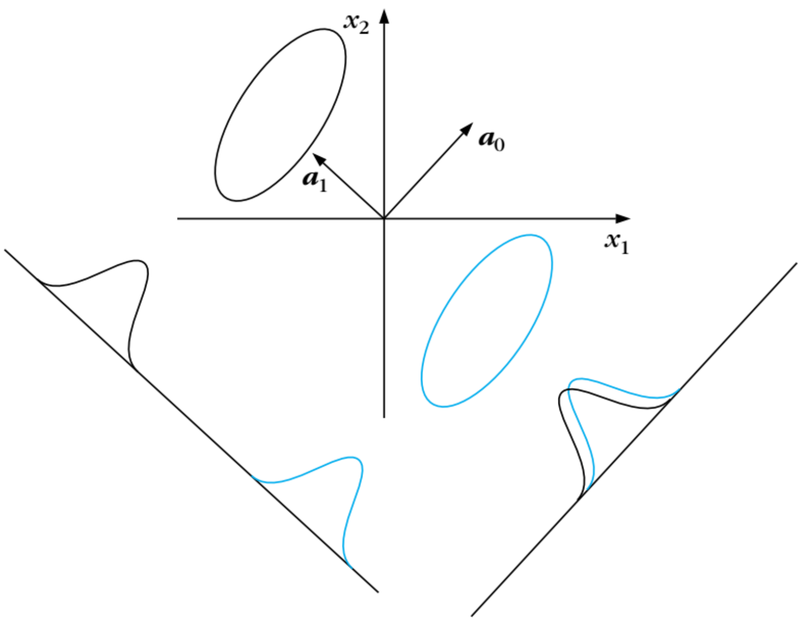
Obs. 1: demais colunas e linhas de **U**, **Σ** e **V** são zero.

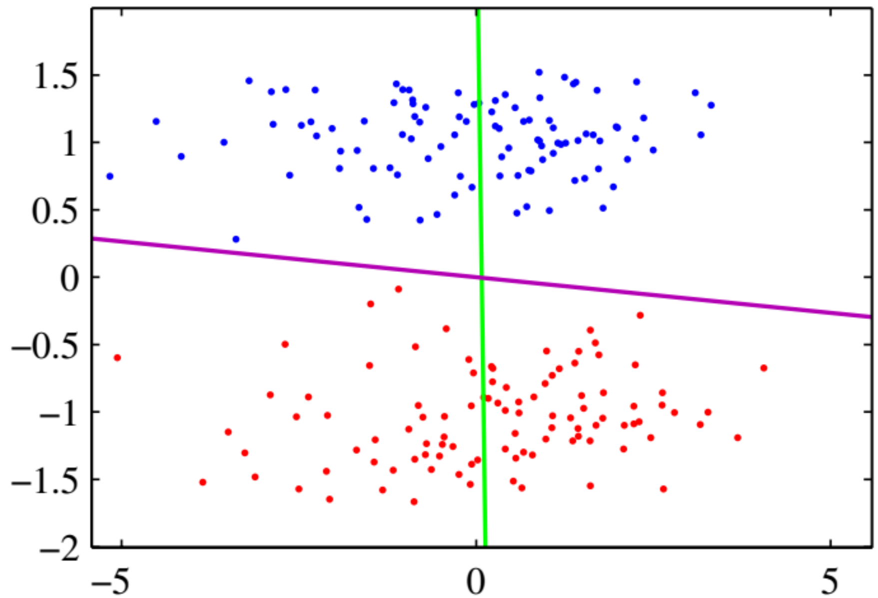
Obs. 2: Desconsiderar valores negativos.

Relação com *Singular Value Decomposition* (SVD)

- Pode-se fazer então, a SVD para obter a *eigendecomposition* de $\mathbf{S} = \mathbf{X}\mathbf{X}^T$: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, sendo:
  - As colunas da matriz **U** são os autovetores não-nulos de $\mathbf{S} = \mathbf{X}\mathbf{X}^T$.
  - A matriz **Σ** é uma matriz diagonal r x r cujos elementos da diagonal são os valores singulares $\sigma_{ii}^2 = \lambda_{ii}$, ou seja: autovalores de $\mathbf{S} = \mathbf{X}\mathbf{X}^T$.
  - As colunas da matriz **V** são os autovetores de $\mathbf{U}^T=\mathbf{X}^T\mathbf{X}$.

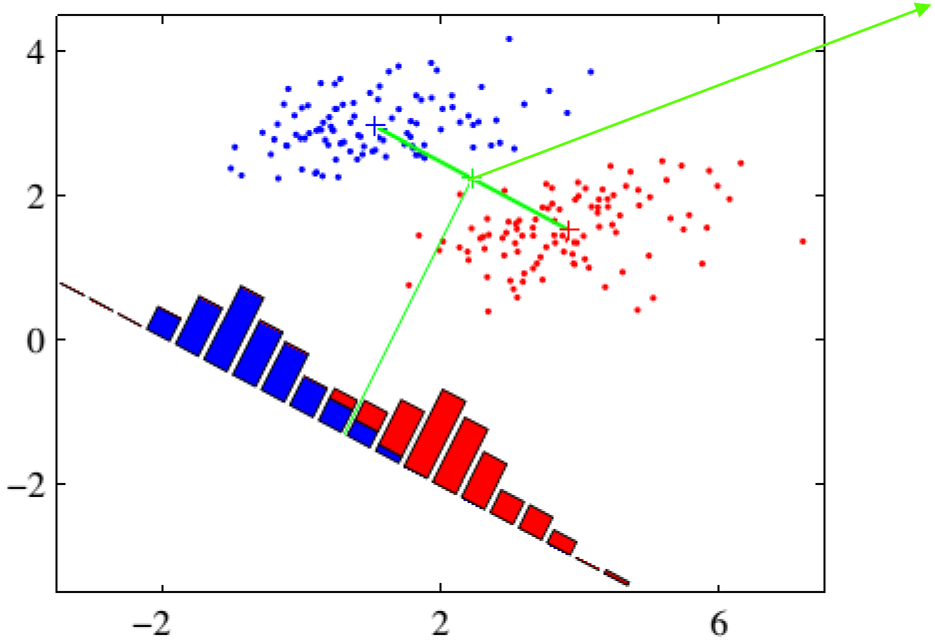
Limitação PCA





Discriminante Linear de Fischer

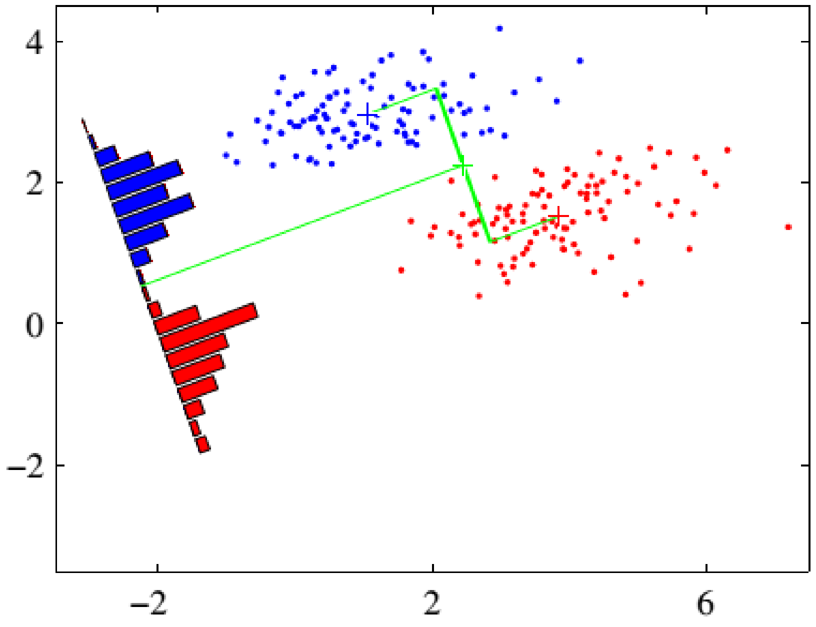
- **Ideia inicial**: projetar (ortogonalmente) os dados (2-D) em uma dimensão que maximize a separação entre as classes, dada, p.ex., pelo vetor definido pelas médias.
- **Problema**: sobreposição (overlapping)! Matrizes de covariância altamente não-diagonais.



$$\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$$

Discriminante Linear de Fischer

- **Alternativa**: realizar a projeção, maximizando a separação entre classes e minimizando a variância de cada classe:



$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}.$$

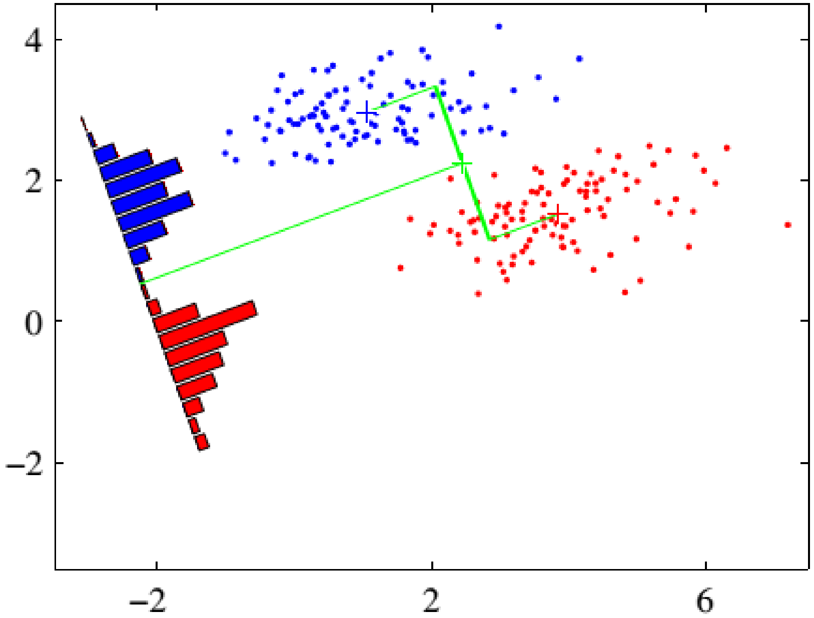
$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2$$

$$y_n = \mathbf{w}^{\mathrm{T}} \mathbf{x}_n.$$

$$m_k = \mathbf{w}^{\mathrm{T}} \mathbf{m}_k$$

Discriminante Linear de Fischer

- **Alternativa**: realizar a projeção, maximizando a separação entre classes e minimizando a variância de cada classe:



$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$$

Matriz de Covariância entre-classes:

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}$$

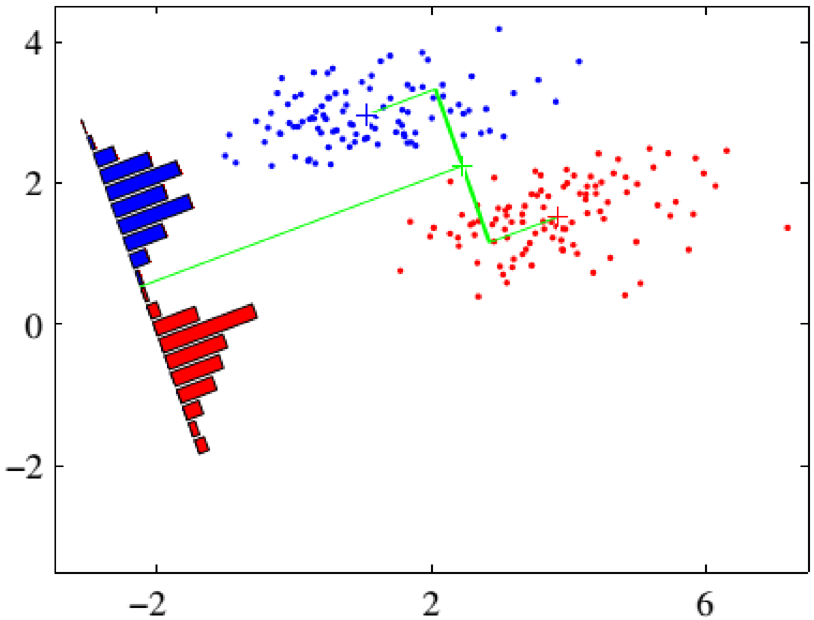
Matriz de Covariância intra-classe:

$$\mathbf{S}_{\mathrm{W}} = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}}$$

Discriminante Linear de Fischer

- Derivando em relação à **w**, obtém-se:

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$$



$$(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})\mathbf{S}_{\mathrm{W}}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})\mathbf{S}_{\mathrm{B}}\mathbf{w}.$$

A magnitude de **w** não é relevante, apenas sua direção. Portanto pode-se: eliminar os seguintes termos (escalares):

$$(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}) \qquad (\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})$$

$\mathbf{S}_{\mathrm{B}}\mathbf{w}$ is always in the direction of $(\mathbf{m}_2 - \mathbf{m}_1)$

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}$$

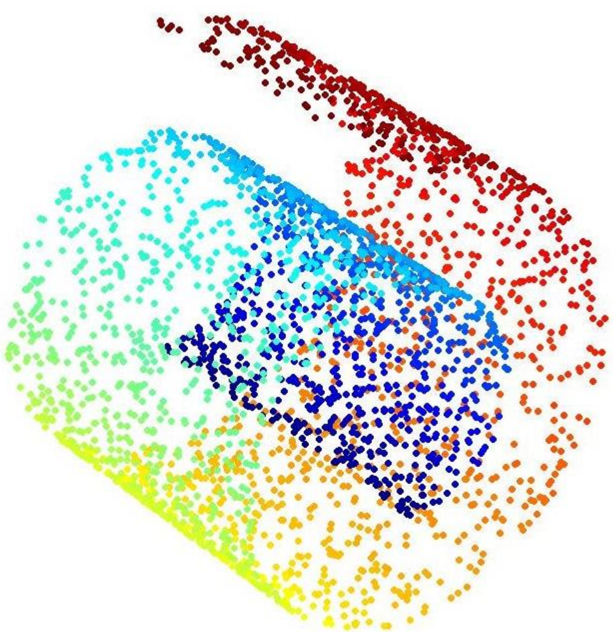
Multiplicando por: $\mathbf{S}_{\mathrm{W}}^{-1}$

Exemplo MATLAB!

Obtém-se que: $\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$

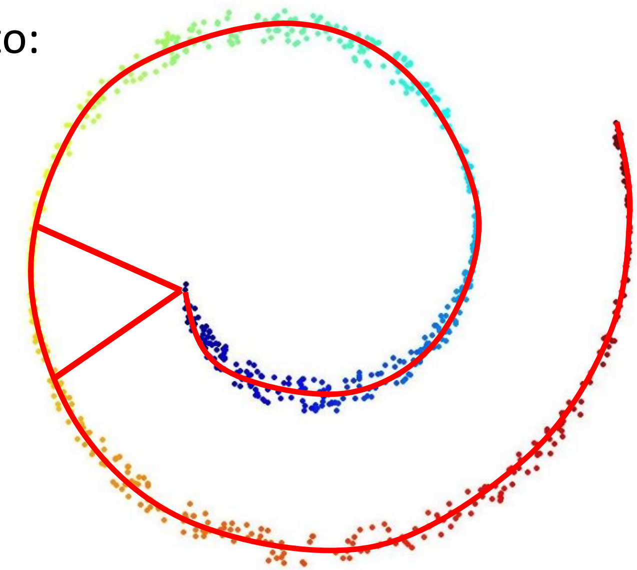
Modelos Não-Lineares

- **Problema**: Se os dados estiverem localizados em um *manifold* curvilíneo, como no conjunto de dados abaixo, a projeção de máxima variância (linear – PCA) resulta em um mapeamento que leva em conta distância euclidiana entre pares de pontos e não as distâncias implícitas do *manifold*.



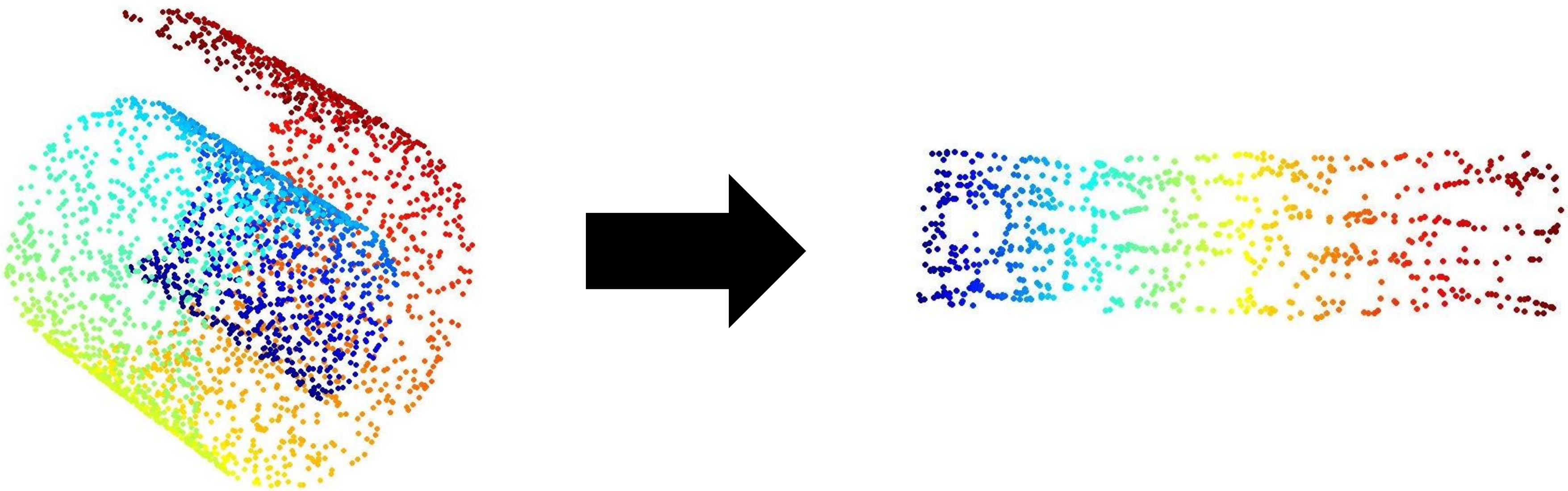
PCA

Preserva as distâncias no mapeamento:



Modelos Não-Lineares

- Ideia geral:



Kernel PCA

- Supondo que:

$$\hat{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\phi}(\boldsymbol{x}_n) \boldsymbol{\phi}(\boldsymbol{x}_n)^T.$$

$$\boldsymbol{x} \in \mathbb{R}^l \longmapsto \boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{H}.$$

- Com isso, pode-se realizar uma decomposição em autovetores e autovalores.
- Essa decomposição é equivalente à decomposição da matriz de Kernel (*Kernel Matrix*) (**PROVA - EXERCÍCIO**):

$$\mathcal{K}(i,j) = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Kernel PCA

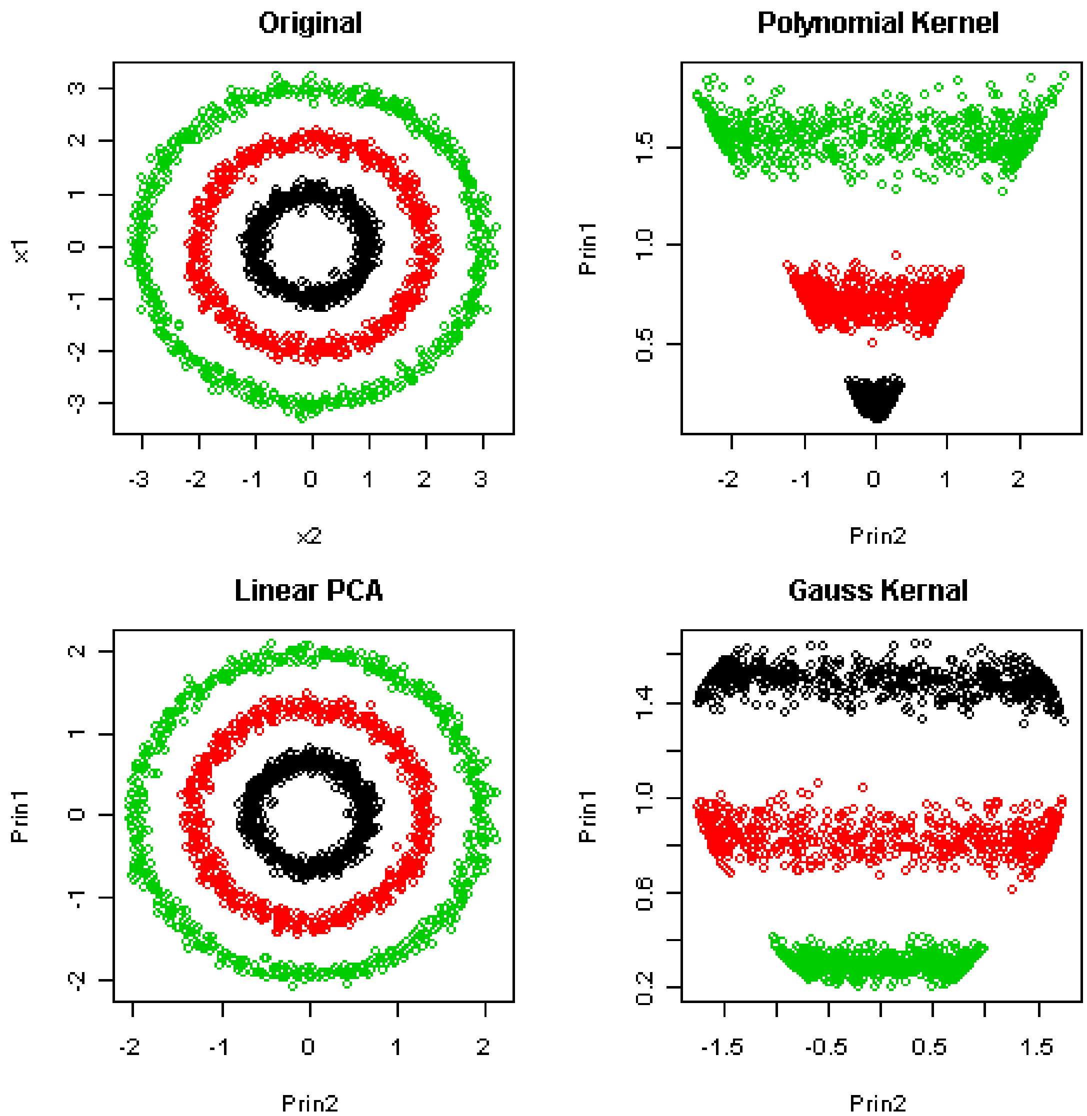
- Calcule a *kernel matrix N×N*, sendo $K(i,j) = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Realize a normalização equivalente a subtrair a média dos dados no espaço de características, definida por:

$$k_{ij} = -\frac{1}{2}\left(k_{ij} - \frac{1}{n}\sum_{l} k_{il} - \frac{1}{n}\sum_{l} k_{jl} + \frac{1}{n^2}\sum_{lm} k_{lm}\right)$$

- Calcule os m autovalores/autovetores dominantes λ_k e $\boldsymbol{a}_k$ de **K** ($k = 1, 2, \ldots, m$). Normalmente, normaliza-se também os autovetores para norma unitária no espaço de características.
- Dado um vetor $\boldsymbol{x} \in R^l$, obtenha sua representação um espaço de dimensão inferior calculando as *m* projeções em relação a cada autovetor, ou seja:

$$z_k = \sum_{n=1}^{N} a_{kn}\kappa(\boldsymbol{x}, \boldsymbol{x}_n),\ k = 1, 2, \ldots, m.$$

Kernel PCA


Original
x1
x2
-3 -2 -1 0 1 2 3
Polynomial Kernel
Prin1
Prin2
0.5 1.0 1.5
-2 -1 0 1 2
Linear PCA
Prin1
Prin2
-2 -1 0 1 2
Gauss Kernal
Prin1
Prin2
0.2 0.6 1.0 1.4
-1.5 -0.5 0.5 1.5

Feature Selection

- Um algoritmo de *feature selection* pode ser visto como a combinação de uma técnica de busca de novos subconjuntos de *features*, juntamente com uma medida de avaliação que classifica e ordena os diferentes subconjuntos de *features*.
- O algoritmo mais simples é testar cada subconjunto possível, encontrando o que minimiza a taxa de erro.
- Busca exaustiva - normalmente intratável.

Feature Selection

- Em geral, utilizam-se métodos de busca alternativos:
  - Simulated annealing;
  - Genetic algorithm;
  - Greedy forward selection;
  - Greedy backward elimination.
- Fogem do escopo da disciplina. Sugestão:

Feature Selection Algorithms: A Survey and Experimental Evaluation

Luis Carlos Molina, Lluís Belanche, Àngela Nebot

Universitat Politècnica de Catalunya

Departament de Llenguatges i Sistemes Informátics

Referências

- Paper: Dimensionality Reduction: A Comparative Review. Laurens van der Maaten, Eric Postma, Jaap van den Herik
- Stanford University – Mining Massive Datasets