Sistemas Inteligentes

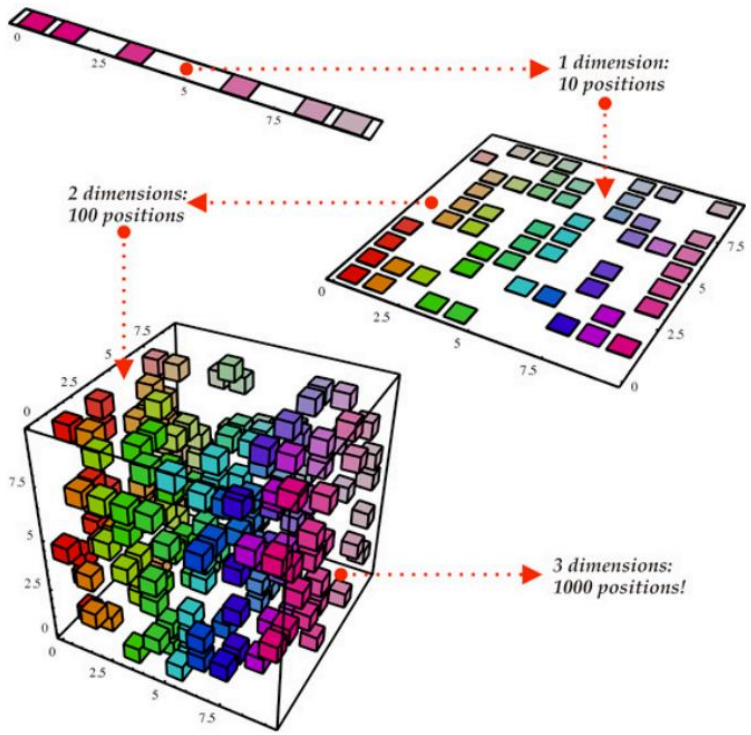
REDUÇÃO DE DIMENSIONALIDADE DOS DADOS

Reduzindo a dimensão dos dados de entrada

Diferentes técnicas podem ser utilizadas para se tentar reduzir a dimensão dos dados que serão utilizados

Alta dimensão dos dados requer um número cada vez maior de dados para o treinamento → perda de desempenho dos sistemas classificadores

Redução, entretanto, não pode ser extrema, pois deve-se preservar o máximo de informação possível



Feature Extraction

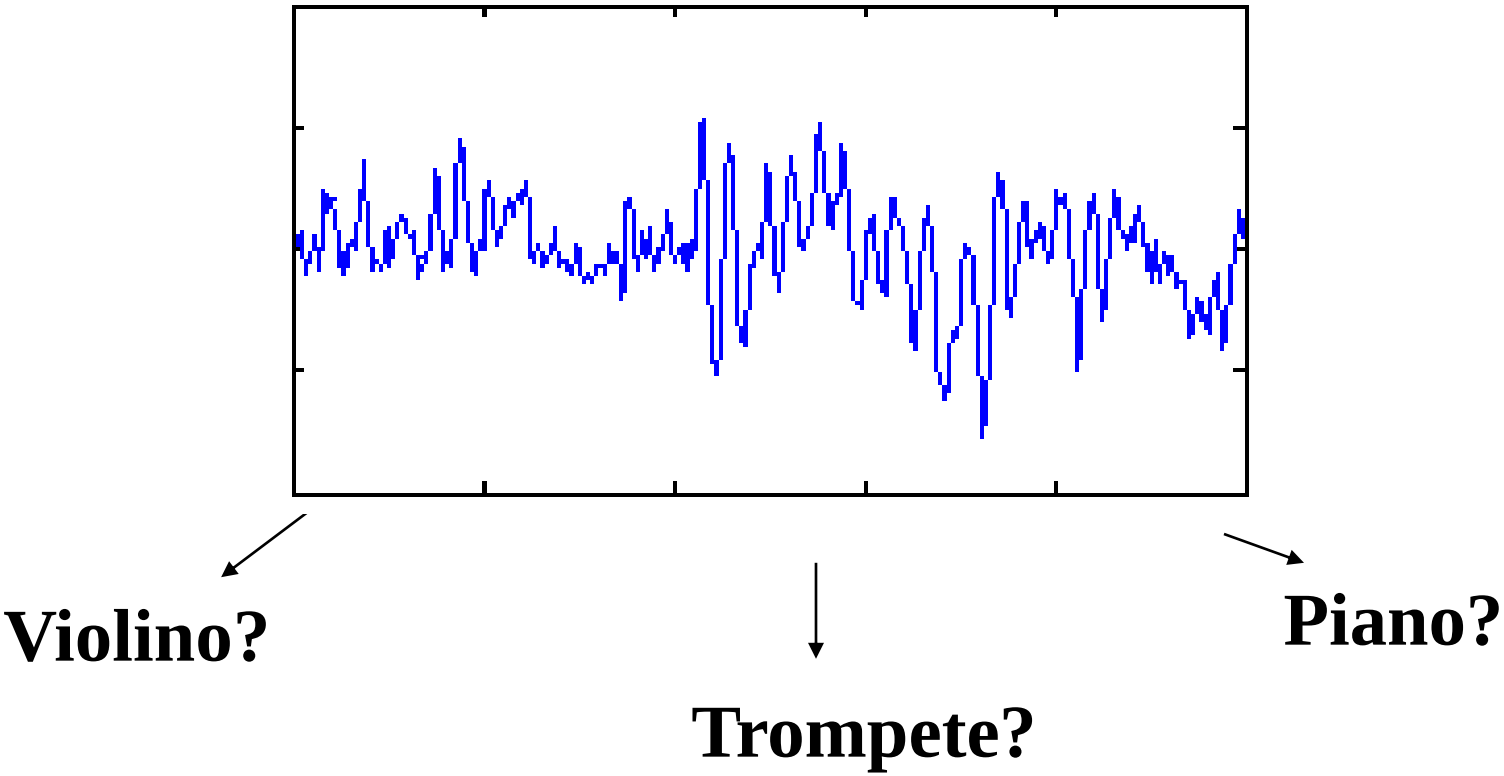
Feature Extraction se refere ao processo de transformação de dados arbitrários – como imagens, sons, texto – em características numéricas que podem ser utilizadas pelas técnicas de classificação, por exemplo.

No exemplo do início do curso, vimos um exemplo de um sistema de reconhecimento de padrões no qual o objetivo era separar automaticamente duas classes de peixes baseado em imagens.

Uma possiblidade seria considerar como *features* os pixels da imagem coletada. Essa abordagem, entretanto, mesmo para imagens de baixa resolução (e.g., 100x100 pixels), geraria dados de entrada (as imagens) com dimensão elevada (nesse exemplo, com um vetor de 100000 elementos!).

Dessa forma, escolher adequadamente as características a serem utilizadas no processo de classificação é essencial e está diretamente relacionado à redução da dimensão dos dados de entrada do classificador.

Exemplo – Identificação de Instrumentos Musicais



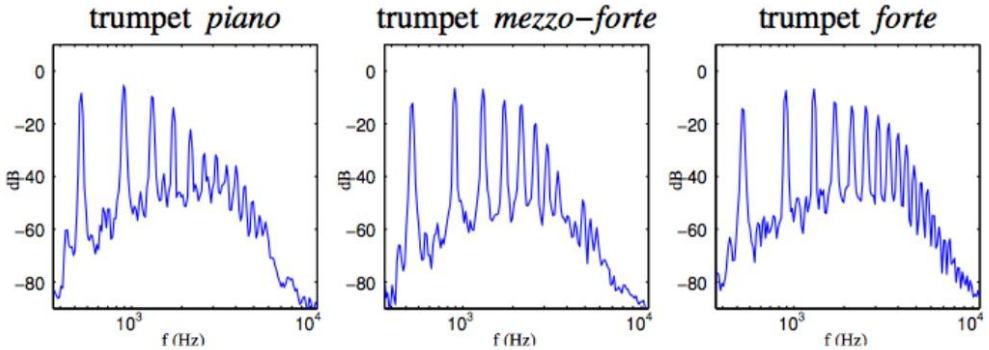
Que tipo de *features* utilizar?

Não há uma resposta única correta para a pergunta – tudo depende da aplicação

Algumas possibilidades

- Características temporais – ritmo ,envelope da amplitude do sinal, etc
- Características frequenciais – espectro, relação entre as harmônicas, etc
- Características tempo-frequenciais – espectrograma, wavelets, MFCC, etc

Espera-se, entretanto, que exista certa variabilidade dos sons dentro de uma mesma classe

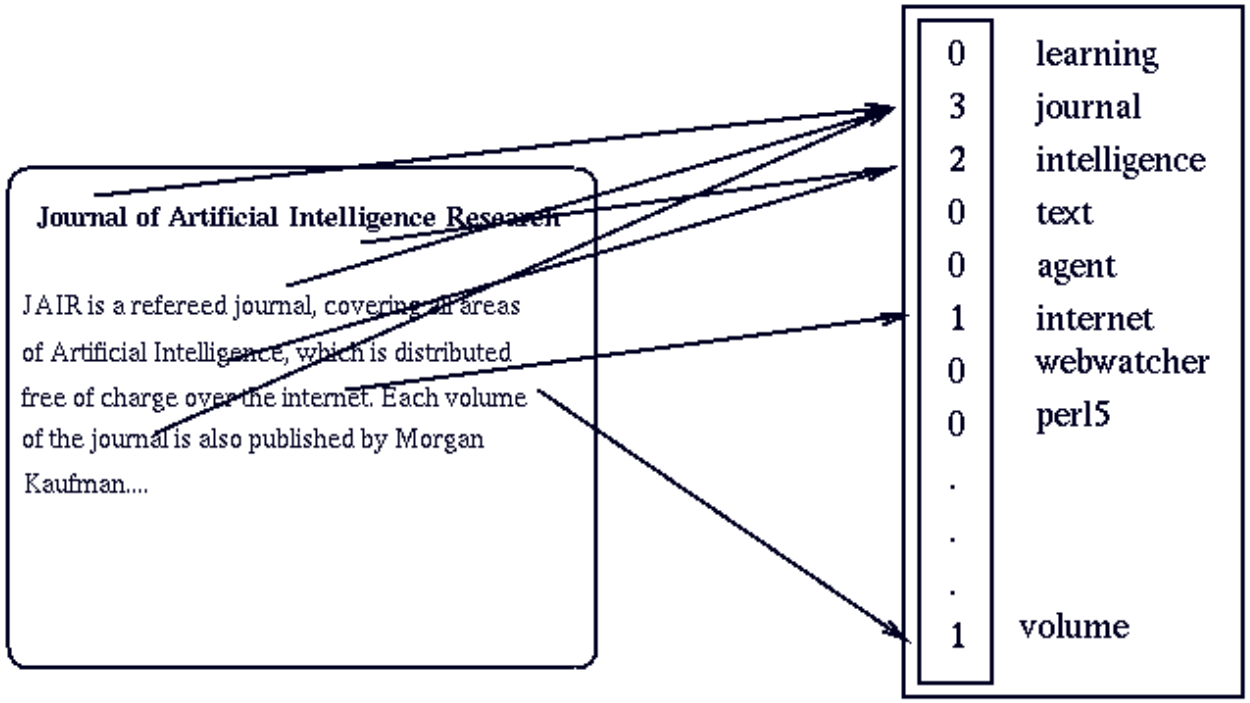


Exemplo - Classificação de Textos com modelo de espaço vetorial

Uma abordagem usual para trabalhar com documentos é represent-los como um vetor numérico (esparso).

- Com isso, ignora-se a estrutura linguística do texto (pode não ser tão interessante em alguns casos)
- A abordagem é normalmente denominada representação “Bag-of-Words” ou “Modelo de Espaço Vetorial”This representation is referred to also as “Bag-Of-Words” or “Vector-Space-Model”
- Modelo pode ser utilizado tanto para classificação como clusterização

Representação Bag-of-Words



Pesos das Palavras

Cada palavra deve ter um peso, que pode dar uma ideia de sua "importância" no texto. Um dos métodos para se determinar esse peso é denominado de frequência normalizada de palavras (term-frequency inverse document frequency - TFIDF), calculada da seguinte forma:

$$\mathbf{x}_d = TFIDF(t,d) = TF(t,d) \times \log \frac{n_d}{DF(d,t)}$$

onde

n_d é o número de documentos

$TF(t,d)$ denota a frequência do termo t no documento d

$DF(d,t)$ é o número de documentos que contêm o termo t

Exemplo de Representação de um texto

TRUMP MAKES BID FOR CONTROL OF RESORTS Casino owner and real estate Donald Trump has offered to acquire all Class B common shares of Resorts International Inc, a spokesman for Trump said. The estate of late Resorts chairman James M. Crosby owns 340,783 of the 752,297 Class B shares. Resorts also has about 6,432,000 Class A common shares outstanding. Each Class B share has 100 times the voting power of a Class A share, giving the Class B stock about 93 pct of Resorts' voting power.

Texto Original

[RESORTS:0.624] [CLASS:0.487] [TRUMP:0.367] [VOTING:0.171]
[ESTATE:0.166] [POWER:0.134] [CROSBY:0.134] [CASINO:0.119]
[DEVELOPER:0.118] [SHARES:0.117] [OWNER:0.102] [DONALD:0.097]
[COMMON:0.093] [GIVING:0.081] [OWNS:0.080] [MAKES:0.078]
[TIMES:0.075] [SHARE:0.072] [JAMES:0.070] [REAL:0.068]
[CONTROL:0.065] [ACQUIRE:0.064] [OFFERED:0.063] [BID:0.063]
[LATE:0.062] [OUTSTANDING:0.056] [SPOKESMAN:0.049]
[CHAIRMAN:0.049] [INTERNATIONAL:0.041] [STOCK:0.035]
[YORK:0.035] [PCT:0.022] [MARCH:0.011]

Bag-of-Words

Como comparar textos (vetores)?

Como cada texto é representado por um vetor $\mathbf{x}$, podemos utilizar ferramentas associadas à álgebra linear para comparar os textos.

Uma das formas usuais emprega a similaridade do cosseno (que, como já vimos, está associada ao produto interno entre vetores)

- Calcula o cosseno do "ângulo entre os documentos"
- O valor absoluto da similaridade varia entre 0 (textos diferentes) e 1 (iguais)

$$sim(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1^T \mathbf{x}_2}{\| \mathbf{x}_1 \| \| \mathbf{x}_2 \|} = \frac{\sum_i \mathbf{x}_1(i) \mathbf{x}_2(i)}{\sqrt{\sum_i \mathbf{x}_1(i)^2} \sqrt{\sum_i \mathbf{x}_2(i)^2}}$$

Quanto mais *features* melhor?

Nem sempre a ideia de quanto maior o número de *features* melhor será o desempenho do sistema classificador é verdadeira.

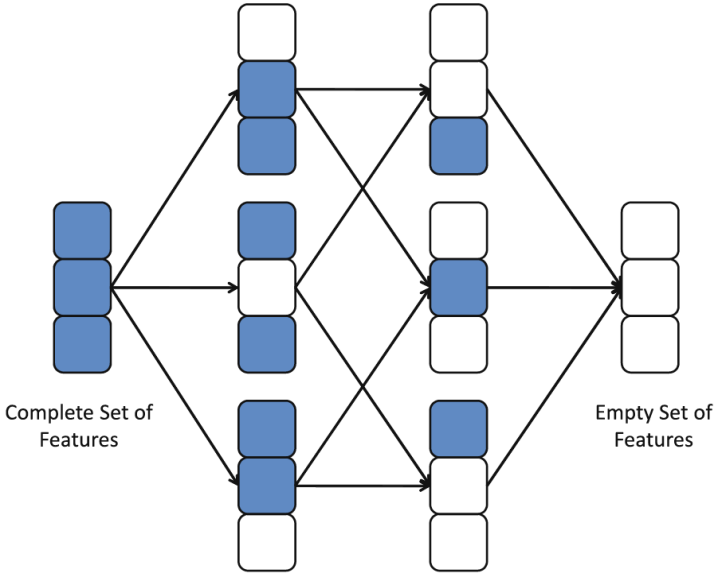
Por outro lado, a redução no número de *features* (e consequente redução da dimensão dos dados de entrada) pode ter consequências bastante interessantes:

- Melhora no desempenho (em termos da velocidade de treinamento, capacidade de predição e simplicidade do modelo)
- Facilita visualização dos dados para a seleção dos modelos
- Redução de ruído

Nesse contexto é que se insere o processo de Seleção de Features, que tem como objetivo escolher um subconjunto ótimo de features de acordo com um critério pré estabelecido.

Procurando um subconjunto de *features*

O processo de seleção de *features* pode ser visto como um problema de busca combinatorial. Considere, para isso, que a seleção é feita por meio de um vetor binário, de dimensão igual ao vetor de *features*, no qual cada elemento com valor 1 indica que a *feature* associada deve ser selecionada, e o valor 0 indica que ela deve ser descartada.



Há, nesse caso, 2^M possíveis combinações, onde M é o número de características total.

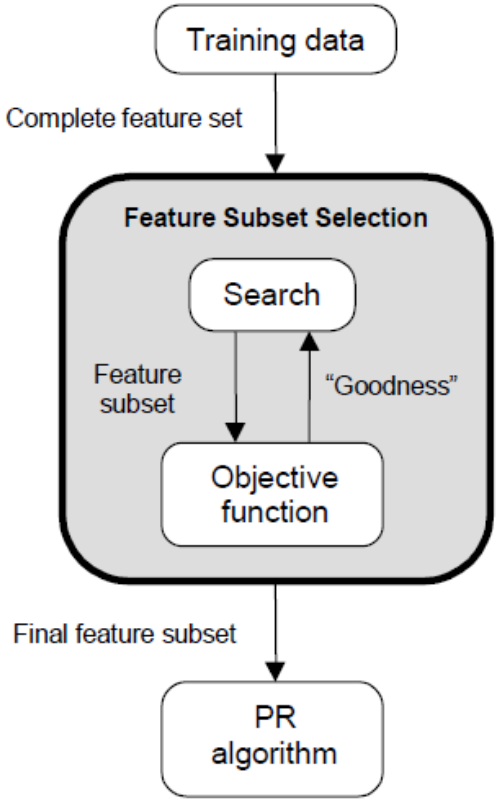
Seleção de Features

Estratégias de **Busca**

- Ótima (Exaustiva)
- Heurística
- Randomizada

Estratégias de **Avaliação**

- Filtragem
- *Wrapper*
- Embedded



Como realizar a busca?

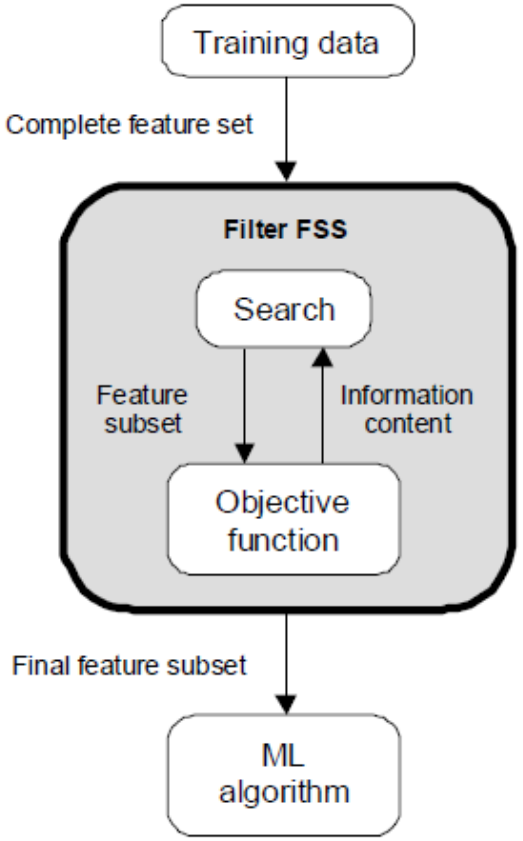
Há diferentes estratégias para tentar obter o subconjunto ótimo de *features*:

- **Sequential Forward Generation (SFG):** Processo inicia com um subconjunto vazio de features e vai adicionando features de acordo com algum critério objetivo que seja capaz de distinguir que uma feature é melhor que a outra, até que um determinado critério de parada seja atingido.
- **Sequential Backward Generation (SBG):** Processo reverso, em que se inicia com todas as features e vai removendo features de acordo com critério que indique qual a pior feature (ou menos importante).
- **Bidirectional Generation (BG):** Realiza a busca em ambas as direções concomitantemente, até que um dos processos atinja seu critério de parada ou tenham chegado à metade do caminho.
- **Random Generation (RG):** realiza a busca de maneira aleatória.
- **Busca exaustiva**: explora todos os possíveis subconjuntos para tentar determinar o subconjunto ótimo
- **Busca heurística**: utiliza algum método heurístico para a busca, evitando assim a busca exaustiva

Estratégias de Avaliação - Filtragem

Essa estratégia independe do algoritmo de classificação, i.e., a seleção das características utiliza métrica que não se baseia diretamente no resultado da classificação

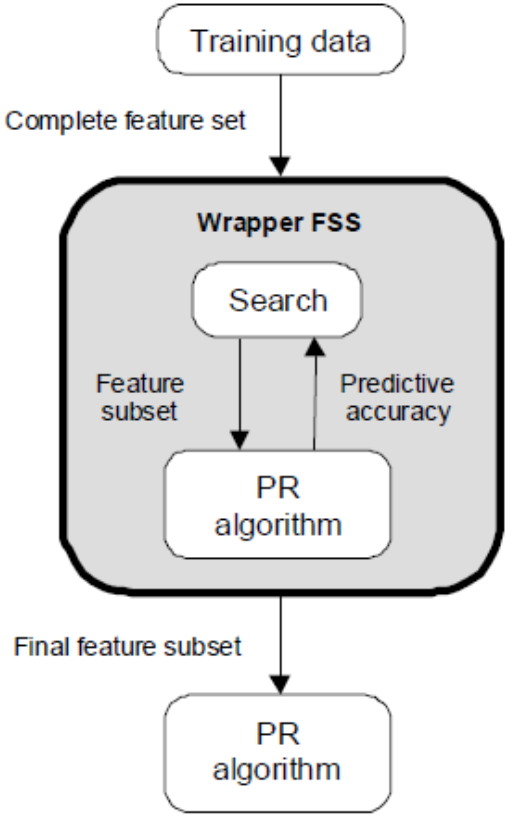
- A função avalia os subconjuntos de *features* com base, por exemplo, na distância entre as classes, dependência estatística ou medidas baseadas na teoria da informação



Estratégias de Avaliação - Wrapper

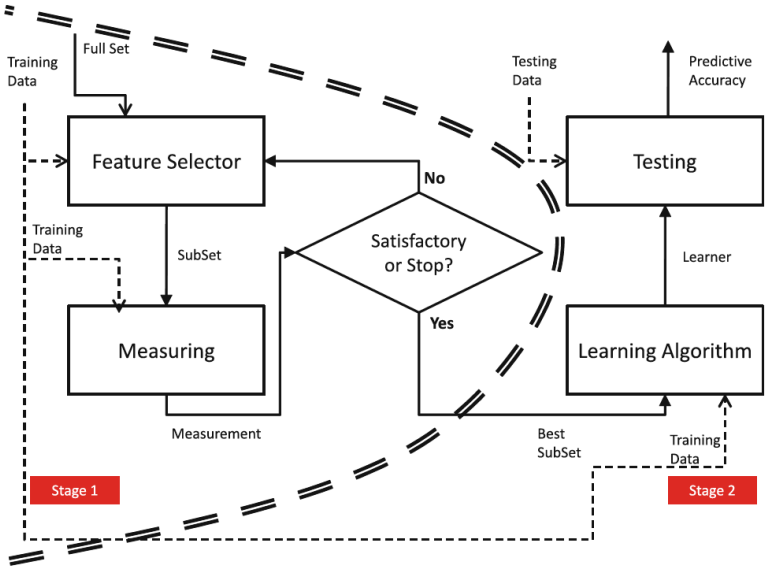
Utiliza o resultado da classificação como critério para a avaliação do subconjunto de *features*

- A função objetivo, portanto, depende de um classificador, que realiza a classificação dos dados utilizando o subconjunto de features.

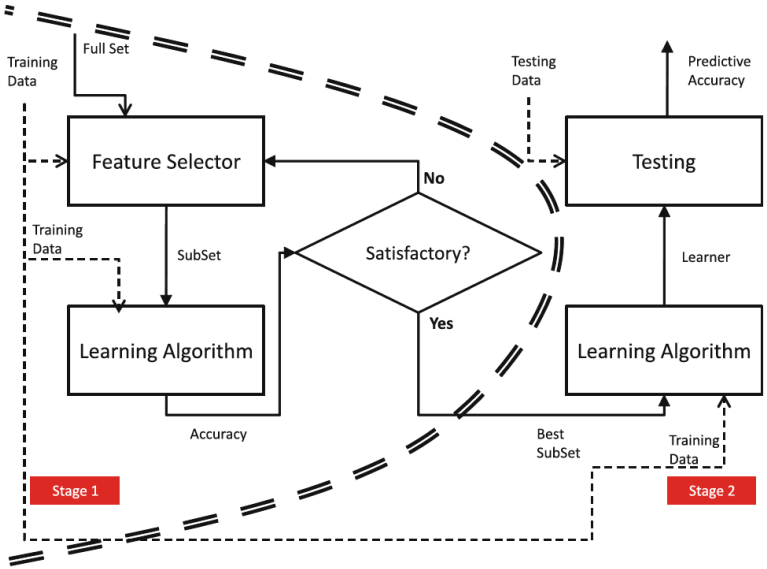


Filtragem x Wrapper

FILTRO



WRAPPER



Métodos gerais para redução de dimensionalidade

Alguns métodos estatísticos podem ser encarados como ferramentas gerais para a redução de dimensionalidade dos dados

- **Principal Component Analysis (PCA)**
- Independent Component Analysis (ICA)
- **Linear Discriminant Analyisis (LDA)**
- Non-negative Matrix Factorization (NMF)

De uma forma ou de outra, a ideia é explorar a redução da dimensionalidade dos dados (projeção em um sub-espaço) de maneira que não sejam perdidas informações relevantes (“compressão com perdas”)

As ferramentas também podem ser usadas para a extração e seleção automática de características

O que é um sub-espaço?

Problema: encontrar uma base para um sub-espaço de menor dimensão

- Aproximar vetores “projetando-os” em um sub-espaço de menor dimensão

$$\mathbf{x} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_N\mathbf{v}_N$$

onde $\mathbf{v}_n$ representam a base do espaço N-dimensional

$$\hat{\mathbf{x}} = b_1\mathbf{u}_1 + b_2\mathbf{u}_2 + \cdots + b_K\mathbf{u}_K$$

K<N. Se K=N, então $\hat{\mathbf{x}} = \mathbf{x}$.

O que é um sub-espaço?

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{(standard basis)}$$

$$x_v = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 3v_1 + 3v_2 + 3v_3$$

$$u_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, u_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{(some other basis)}$$

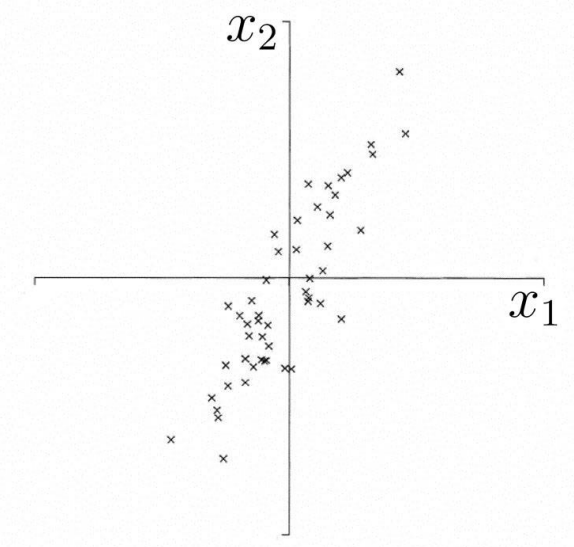
$$x_u = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 0u_1 + 0u_2 + 3u_3$$

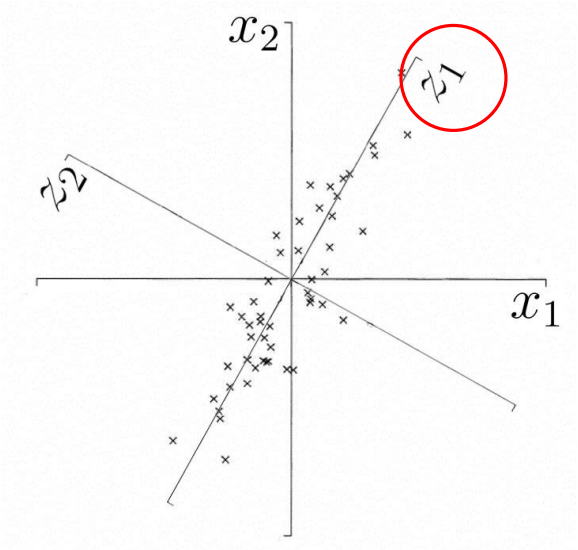
thus, $x_v = x_u$

Análise por Componentes Principais

Motivação

- Encontrar bases que contenham alta variância dos dados
- Codificar dados com baixo erro de representação (baixo MSE)





Análise por Componentes Principais (PCA)

Suponha que

$$E\{\mathbf{x}\} = 0$$

E considere a projeção dos dados em uma direção específica $\mathbf{q}$ (com $\| \mathbf{q} \| = 1$), i.e.

$$y = \mathbf{q}^T\mathbf{x} = \mathbf{x}^T\mathbf{q}$$

O objetivo da PCA é encontrar as direções em que as projeções apresentem a máxima variância, i.e.,

$$E\{y^2\} = E\{(\mathbf{q}^T\mathbf{x})(\mathbf{x}^T\mathbf{q})\} = \mathbf{q}^T\mathbf{R}\mathbf{q}$$

onde $\mathbf{R}$ denota a matriz de autocorrelação dos dados – contém as correlações entre as componentes dos vetores $\mathbf{x}$.

Análise por Componentes Principais (PCA)

Solução é dada pelos Autovalores e Autovetores de **R**

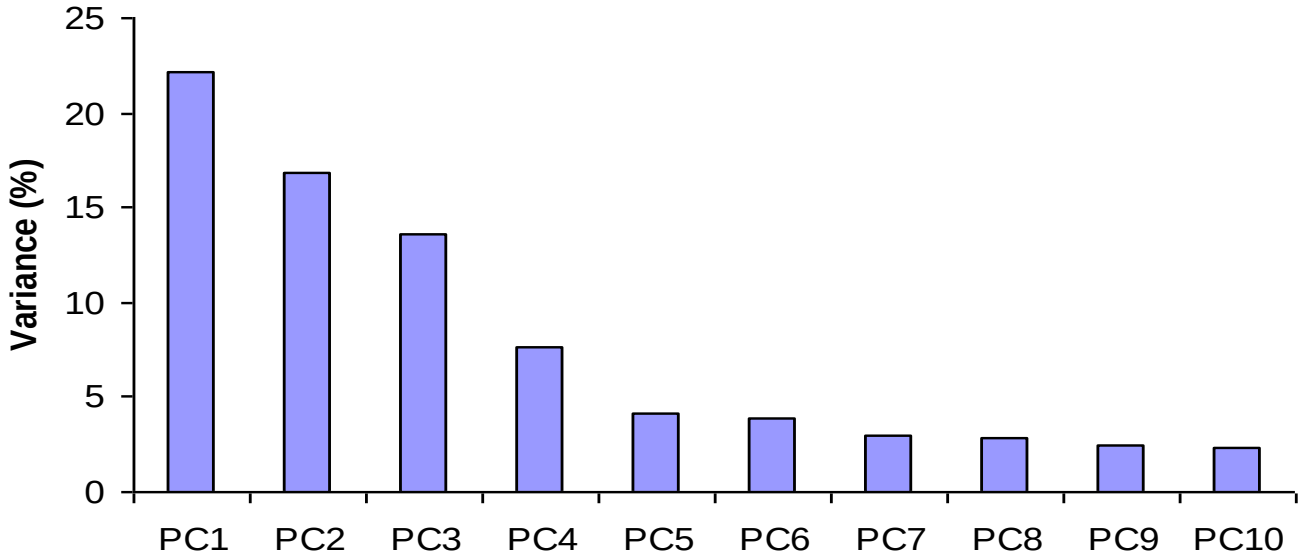
$$\mathbf{R} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T, \quad \mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_j, ..., \mathbf{q}_m], \boldsymbol{\Lambda} = diag[\lambda_1, \lambda_2, ..., \lambda_j, ..., \lambda_m]$$

$$\Leftrightarrow \mathbf{R}\mathbf{q}_j = \lambda_j \mathbf{q}_j \quad j = 1, 2, ..., m$$

Direções principais → associadas aos autovetores

Variância das projeções nas direções principais → autovalores

Redução da Dimensionalidade



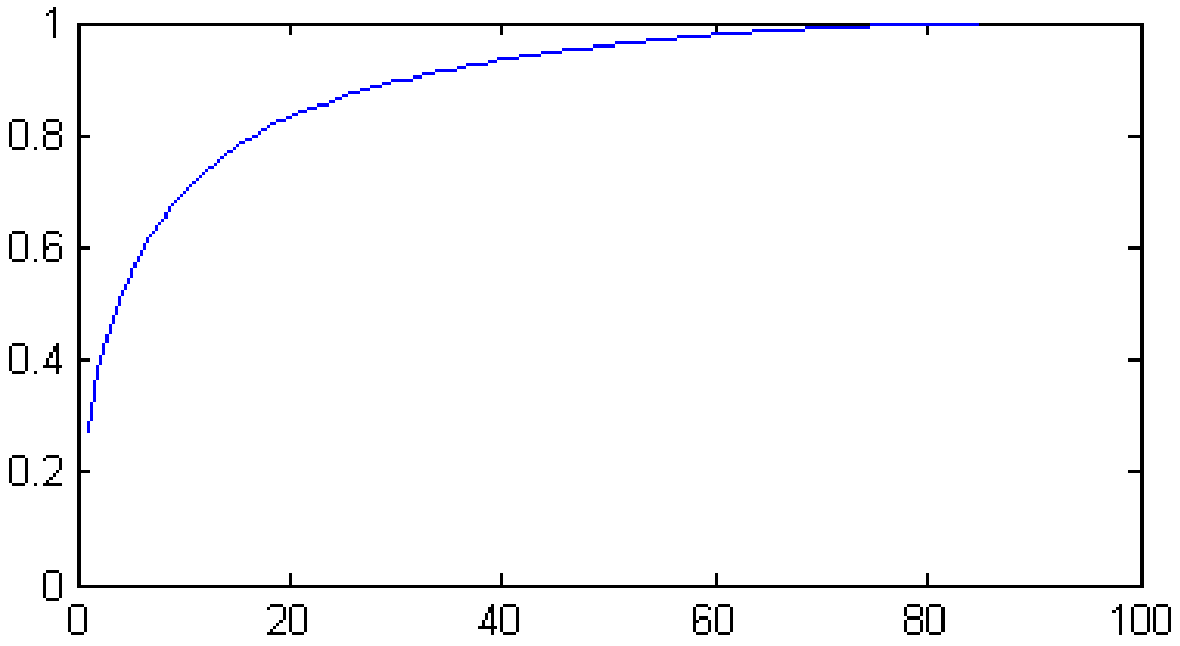
Pode-se ignorar componentes com menos significância.

Perde-se alguma informação, mas se os autovalores são pequenos, não se perde muita informação

- *n* dimensões nos dados originais
- Calcule os *n autovalores e autovetores*
- Escolha os primeiros *p* autovetores, baseados nos seus autovalores
- Dado final conterá apenas *p* dimensões.

Redução de Dimensionalidade

Variância



Dimensão

Reconstrução usando as Componentes Principais

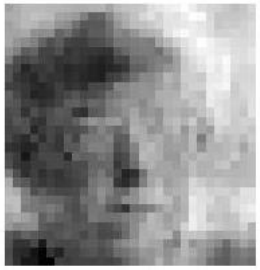
q=1

q=2

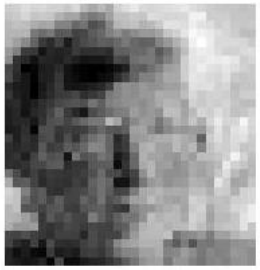
q=4

q=8

q=16



q=32



q=64

q=100…

Imagem Original

LDA como ferramenta de redução de dimensionalidade

Conforme foi visto anteriormente, a LDA realiza a projeção dos dados em uma direção que maximiza a separação das classes

Assim, é possível utilizar o mesmo procedimento para projetar os dados em um subespaço (não necessariamente 1D) no qual as classes apresentam a máxima separação. Dessa forma, a tarefa de classificação pode se tornar mais simples.