Aprendizagem Automática

Aula Prática

Geração de Dados Aleatórios e Representações, Visualizações e Transformações de Dados Univariados e Multivariados

𝔾. Marques

Variáveis Aleatórios

A capacidade de gerar variáveis aleatórias (v.a) é uma componente fundamental no processo de desenvolvimento e simulação de métodos de aprendizagem automática. O sub-módulo `random` do `NumPy` providencia várias rotinas de geração de números pseudo-aleatórios[1].

Algumas funções do módulo `np.random`:

• `rand()`	dist. uniforme $[0, 1]$	• `laplace()`	dist. de Laplace
• `randn()`	gaussiana $\mathcal{N}(0, 1)$	• `wald()`	dist. de Wald
• `randint()`	inteiros aleatórios	• `triangular()`	dist. de triangular
• `rayleigh()`	dist. de Rayleigh	• `binomial()`	dist. binomial
• `cauchy()`	dist. de Cauchy	• `poisson()`	dist. de Poisson
• `logistic()`	dist. logística	• `geometric()`	dist. geométrica
• `exponential()`	dist. exponencial	• `permutation()`	permutações

• `seed()` Inicialização do gerador de números aleatórios. Importante em simulação para, por exemplo, gerar os mesmos dados, parâmetros, etc.

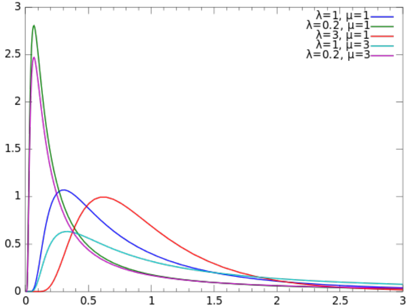
[1] Diz-se pseudo-aleatórios porque os números são gerados de forma determinística mas a sua distribuição assemelha-se à de números gerados aleatoriamente.

Variáveis Aleatórias

Exemplo 1: Geração de variáveis aleatórias 1D (`L1AAex001.py`)

- Objectivo: Gerar pontos 1D segundo a distribuição de Wald (também denominada distribuição gaussiana inversa):

$$p(x|\mu,\lambda) = \left(\frac{\lambda}{2\pi x^3}\right)^{\frac{1}{2}} \exp\left(\frac{-\lambda(x-\mu)^2}{2\mu^2 x}\right), \text{ com } \lambda,\mu > 0 \text{ e } x \in [0,+\infty]$$



Distribuição de Wald para vários valores de μ e λ
(de Krishnavedala via Wikimedia Commons)

Variáveis Aleatórias

Exemplo 1: Geração de variáveis aleatórias 1D (L1AAex001.py)

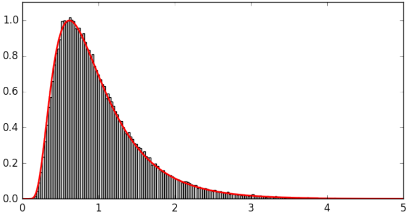
- Objectivo: Gerar pontos 1D segundo a distribuição de Wald (também denominada distribuição gaussiana inversa):

```python
# -*- coding: latin-1 -*-
# geração de variáveis aleatórias 1D
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt
x=rd.wald(1,3,int(100000)) #1e5 pts aleatórios (mu=1,lambda=3)
#fazer histograma + plot da distribuição
hx,b=np.histogram(x,np.linspace(0,5,201),density=True)
#b->array de 201 entradas (fronteiras de quantificação)
#obter valores de quantificação (valor a meio dos intervalos)
b=(b[:-1]+b[1:])/2.0
t=np.linspace(0+1e-6,5,100)
fx=np.sqrt(3/(2*np.pi*t**3))*np.exp(-3*(t-1)**2/(2*t))
plt.figure(figsize=(8,4))
plt.bar(b,hx,width=0.025,color=[0.7,.7,.7])
plt.plot(t,fx,'r',linewidth=2)
#plt.axis([0,5,0,1.1])
plt.show()
```

Variáveis Aleatórias

Exemplo 1: Geração de variáveis aleatórias 1D (`L1AAex001.py`)

- Objectivo: Gerar pontos 1D segundo a distribuição de Wald (também denominada distribuição gaussiana inversa):



Histograma + distribuição de Wald com $\mu = 1$ e $\lambda = 3$

Dados Aleatórios

Exemplo 2: Geração de variáveis aleatórias 1D (`L1AAex002.py`)

- Objectivo: Gerar variáveis 1D uniformemente distribuídas entre $[0,1]$ e verificar a distribuição resultante da soma de N destas variáveis.

Revisão: **Teorema do limite central:**
A distribuição resultante de uma soma de um número suficientemente elevado de variáveis independentes tende para a distribuição gaussiana (normal), independentemente das distribuições subjacentes.

Revisão: **Média e variância de variáveis uniformes:**

$$\text{distribuição:} \quad p(x) = \frac{1}{b-a} \text{ com } x \in [a, b]$$

$$\text{média:} \quad \mu = \mathbb{E}\{x\} = \int_a^b \frac{x}{b-a} dx = \frac{1}{2}(a+b)$$

$$\text{variância:} \quad \sigma^2 = \mathbb{E}\{(x-\mu)^2\} = \int_a^b \frac{(x-\mu)^2}{b-a} dx = \frac{1}{12}(b-a)^2$$

Revisão: $\mathbb{E}\{\cdot\}$ é o operador valor esperado. Ex: Média$\longrightarrow \mathbb{E}\{x\} = \int_{-\infty}^{+\infty} xp(x)dx$

Dados Aleatórios

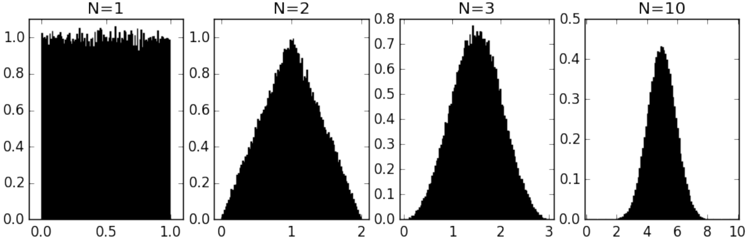
Exemplo 2: Geração de variáveis aleatórias 1D (`L1AAex002.py`)

```python
# -*- coding: latin-1 -*-
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt
plt.close('all')
N=100000 #1e5 pts uniforme
x1=rd.rand(N)
x2=np.sum(rd.rand(2,N),axis=0) #soma de 2 v.a. uniformes
x3=np.sum(rd.rand(3,N),axis=0) #soma de 3 v.a. uniformes
x4=np.sum(rd.rand(10,N),axis=0) #soma de 10 v.a. uniformes
hx,b=np.histogram(x1,np.linspace(0,1,101),density=True)
b=(b[:-1]+b[1:])/2
plt.figure(figsize=(9,2.5))
plt.subplot(141);plt.axis([-0.1,1.1,0,1.1]);plt.title('N=1')
plt.bar(b[0:100],hx,width=0.005);plt.xticks([0,0.5,1])
hx,b=np.histogram(x2,np.linspace(0,2,101),density=True)
b=(b[:-1]+b[1:])/2
plt.subplot(142);plt.axis([-0.1,2.1,0,1.1]);plt.title('N=2')
plt.bar(b[0:100],hx,width=0.01);plt.xticks(np.arange(3))
hx,b=np.histogram(x3,np.linspace(0,3,101),density=True)
b=(b[:-1]+b[1:])/2
plt.subplot(143);plt.axis([-0.1,3.1,0,.8]);plt.title('N=3')
```

Dados Aleatórios

Exemplo 2: Geração de variáveis aleatórias 1D (`L1AAex002.py`)

- Objectivo: Gerar variáveis 1D uniformemente distribuídas entre $[0, 1]$ e verificar a distribuição resultante da soma de N destas variáveis.



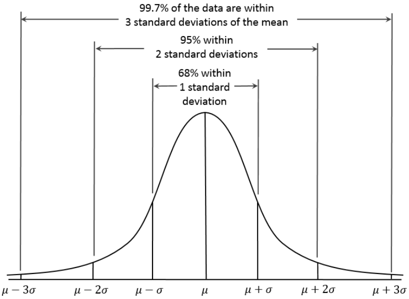
Histograma da soma de v.a. uniformes

- Neste casos todas as variáveis são **independentes e identicamente distribuídas (iid)** com $\mu = \frac{1}{2}$ e $\sigma^2 = \frac{1}{12}$.
- Prova-se que a média e a variância da soma de N destas variáveis são: $\mu_{\text{Total}} = N\mu = \dfrac{N}{2}$ e $\sigma^2_{\text{Total}} = N\sigma^2 = \dfrac{N}{12}$

Dados Aleatórios Gaussianos

A distribuição gaussiana, também denominada distribuição normal, tem um papel preponderante em inferência estatística e na construção de modelos probabilísticos de dados. É habitual usar esta distribuição (ou uma mistura destas distribuições) para modelar dados cujo se desconhece a densidade.

Revisão: Variáveis gaussianas 1D: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$



Distribuição gaussiana 1d
(de Dan Kernler via Wikimedia Commons)

Dados Aleatórios Gaussianos

A distribuição gaussiana, também denominada distribuição normal, tem um papel preponderante em inferência estatística e na construção de modelos probabilísticos de dados. É habitual usar esta distribuição (ou uma mistura destas distribuições) para modelar dados cujo se desconhece a densidade.

- Variáveis gaussianas a d dimensões (notação alternativa: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$)

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

  - $\boldsymbol{\mu}$: vector de média $d \times 1$
  - $\boldsymbol{\Sigma}$: matriz de covariância $d \times d$

Dados Aleatórios Gaussianos

A distribuição gaussiana, também denominada distribuição normal, tem um papel preponderante em inferência estatística e na construção de modelos probabilísticos de dados. É habitual usar esta distribuição (ou uma mistura destas distribuições) para modelar dados cujo se desconhece a densidade.

- Variáveis gaussianas a d dimensões (notação alternativa: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$)

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \quad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \mathrm{cov}(x_1, x_2) & \cdots & \mathrm{cov}(x_1, x_d) \\ \mathrm{cov}(x_2, x_1) & \sigma_2^2 & & \mathrm{cov}(x_2, x_d) \\ \vdots & \cdots & \ddots & \\ \mathrm{cov}(x_d, x_1) & \cdots & & \sigma_d^2 \end{bmatrix}$$

$$\begin{aligned} \mathrm{cov}(x_i, x_j) &= \mathbb{E}\left\{(x_i - \mu_i)(x_j - \mu_j)\right\} \\ \mathrm{cov}(x_i, x_j) &= \mathrm{cov}(x_j, x_i) \\ \mathrm{cov}(x_i, x_i) &= \sigma_i^2 \end{aligned}$$

Dados Aleatórios Gaussianos

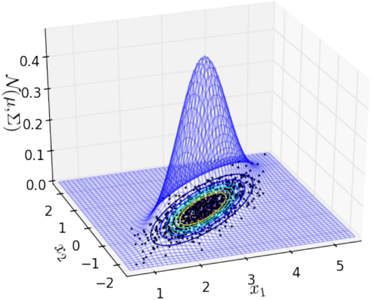
A distribuição gaussiana, também denominada distribuição normal, tem um papel preponderante em inferência estatística e na construção de modelos probabilísticos de dados. É habitual usar esta distribuição (ou uma mistura destas distribuições) para modelar dados cujo se desconhece a densidade.

- Variáveis gaussianas a *d* dimensões (notação alternativa: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$)

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

  - 1000 pts gerados segundo $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, com

$$\boldsymbol{\mu} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix}$$



Dados Aleatórios Gaussianos

A distribuição gaussiana, também denominada distribuição normal, tem um papel preponderante em inferência estatística e na construção de modelos probabilísticos de dados. É habitual usar esta distribuição (ou uma mistura destas distribuições) para modelar dados cujo se desconhece a densidade.

- Propriedades:
  - Definidas únicamente por 2 parâmetros:
    - $\boldsymbol{\mu}$ vector de média
    - $\boldsymbol{\Sigma}$ matriz de covariância
  - Distribuições marginais de v.a. conjuntamente gaussianas são também gaussianas
  - Variáveis resultante de transformações lineares (ex: soma de 2 ou + v.a.) de v.a. gaussianas têm distribuição gaussiana.
  - Variáveis aleatórias guassianas descorrelacionadas são independentes (não acontece com outras distribuições)

Revisão:
  - Variáveis aleatórias independentes são descorrelacionadas.

Representação Matricial de Dados

A representação matricial de dados é essencial no contexto de aprendizagem automática. Na maioria das aplicações, cada objecto (imagens de faces, dígitos, sinais audio, fala, músicas, texto, ...) é representado vectorialmente por um conjunto finito de características (valores). Por sua vez, todos os vectores são "empacotados" numa só matriz de dados. Esta representação é útil em vários aspectos.

- Tira partido das capacidades computacionais associadas a `np.arrays`.
- Torna fácil processar um conjunto de dados em poucos comandos.
- **As instruções em código assemelham-se às equações matriciais de cálculo analítico**.

Representação Matricial de Dados

Abordagem:

- Construir a matriz de dados, $\mathbf{X}$, com $d \times N$ elementos
  - d - nº de elementos de cada vector (dimensão do espaço de características).
  - N - nº de vectores (nº exemplos disponíveis).

Notação:

- Variáveis e escalares - letras minúsculas. Ex: x, δ, ω.
- Vectores - letras minúsculas a carregado. Ex: $\mathbf{x}$, $\boldsymbol{\delta}$, $\boldsymbol{\omega}$.
- Matrizes - letras maiúsculas a carregado. Ex: $\mathbf{X}$, $\boldsymbol{\Delta}$, $\boldsymbol{\Omega}$

  - ou só letras maiúsculas. Ex: X, Δ, Ω

Representação Matricial de Dados

Abordagem:

- Construir a matriz de dados, $\mathbf{X}$, com $d \times N$ elementos

$$\underset{d\times N}{\mathbf{X}} = \left[\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \cdots \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \right] = \left[\begin{bmatrix} x_1[1] \\ x_2[1] \\ \vdots \\ x_d[1] \end{bmatrix} \begin{bmatrix} x_1[2] \\ x_2[2] \\ \vdots \\ x_d[2] \end{bmatrix} \cdots \begin{bmatrix} x_1[N] \\ x_2[N] \\ \vdots \\ x_d[N] \end{bmatrix} \right]$$

$$\begin{array}{ccccc} & 1 & 2 & \cdots & \mathrm{N} \\ = & \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{array}$$

Representação Matricial de Dados

Exemplo 3: geração de v.a. gaussianas 2D (`L1AAex003.py`)

1. Seja $\mathbf{x}$ uma v.a. 2D distribuída segundo $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Gerar uma matriz, $\mathbf{X}$ de 2×1000 com 1000 realizações de $\mathbf{x}$.
2. Transformar $\mathbf{x}$ segundo: $\mathbf{y} = A\mathbf{x} + \mathbf{b}$

$$\text{com} \quad A = \begin{bmatrix} \frac{1}{3\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{3\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad \text{e} \quad \mathbf{b} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

3. A transformação pode ser feita para todo o conjunto: $\mathbf{Y} = A\mathbf{X} + \mathbf{b}$
4. Visualizar os conjuntos $\mathbf{X}$ e $\mathbf{Y}$

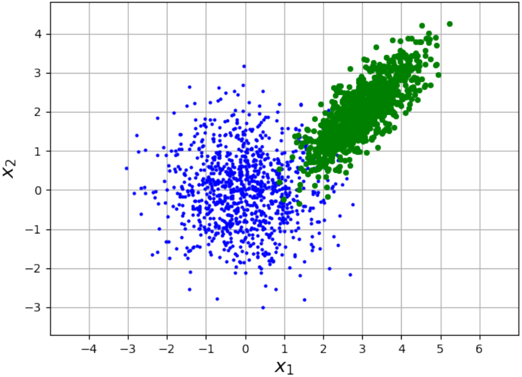
Representação Matricial de Dados

Exemplo 3: geração de v.a. gaussianas 2D (`L1AAex003.py`)

```
# -*- coding: latin-1 -*-
import numpy as np
from matplotlib import pyplot as plt
#criar matrix A 2x2 e vector b 2x1
A=np.array([[1./3/np.sqrt(2.),1./np.sqrt(2.)],\
[-1./np.sqrt(2.)/3,1./np.sqrt(2.)]])
b=np.array([3,2])
#criar 1000 pontos
np.random.seed(0);X=np.random.randn(2,1000)
#para somar b, tem que se transpor x (e voltar a transpor)
Y=(np.dot(A,X).T+b).T
#ou em alternativa: Y=np.dot(A,X)+b[:,np.newaxis]
plt.figure(figsize=(7,5)) #criar figura
plt.plot(X[0,:],X[1,:],'.b',markersize=4)
plt.plot(Y[0,:],Y[1,:],'og',markersize=4)
plt.axis('equal');plt.axis([-5.,7,-4.,5.1])
plt.grid();plt.xticks(np.arange(-4,7))
plt.xlabel('$x_1$',fontsize=16)
plt.ylabel('$x_2$',fontsize=16)
#plt.savefig('../figs/L1AAex003.png',\
#box_inches='tight',transparent=True) #guardar em ficheiro ".png"
plt.show()
```

Representação Matricial de Dados

Exemplo 3: geração de v.a. gaussianas 2D (`L1AAex003.py`)



Representação Matricial de Dados

Exemplo 3 (continuação):

Estimar matriz de covariância, $\mathbf{\Sigma_y}$, de $\mathbf{Y}$.

1. Com a função `np.cov()`:

```
>>> np.cov(Y)
    array([[ 0.51326778, 0.4149577 ],
           [ 0.4149577 , 0.53336087]])
```

2. Através dum produto matricial:

$$\mathbf{\Sigma_y} = \mathbb{E}\left\{(\mathbf{y}-\boldsymbol{\mu}_\mathbf{y})(\mathbf{y}-\boldsymbol{\mu}_\mathbf{y})^\top\right\} \approx \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{y}_n-\boldsymbol{\mu}_\mathbf{y})(\mathbf{y}_n-\boldsymbol{\mu}_\mathbf{y})^\top$$

```
>>> Yn=(Y.T-np.mean(Y,1)).T
>>> np.dot(Yn,Yn.T)/999.  #somatório feito com um só comando!
    array([[ 0.51326778, 0.4149577 ],
           [ 0.4149577 , 0.53336087]])
```

Representação Matricial de Dados

Exemplo 3 (continuação):

Estimar matriz de covariância, $\mathbf{\Sigma_y}$, de **Y**.

2. Através dum produto matricial:

   - Notar que: $\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top \quad \Leftrightarrow \quad$ `np.dot(X,X.T)`
     (daí se poder também fazer `np.dot(Yn,Yn.T)`)

$$\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top = \sum_{n} \underbrace{\begin{bmatrix} x_1[n] \\ x_2[n] \\ \vdots \\ x_d[n] \end{bmatrix}}_{\mathbf{x}_n} \underbrace{\begin{bmatrix} x_1[n] & x_2[n] & \cdots & x_d[n] \end{bmatrix}}_{\mathbf{x}_n^\top}$$

Representação Matricial de Dados

Exemplo 3 (continuação):

Estimar matriz de covariância, $\boldsymbol{\Sigma}_{\mathbf{y}}$, de $\mathbf{Y}$.

2. Através dum produto matricial:

$$\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top = \sum_n \begin{bmatrix} x_1^2[n] & x_1[n]x_2[n] & \cdots & x_1[n]x_d[n] \\ x_1[n]x_2[n] & x_2^2[n] & \cdots & x_2[n]x_d[n] \\ \vdots & & \ddots & \\ x_d[n]x_1[n] & x_d[n]x_2[n] & \cdots & x_d^2[n] \end{bmatrix} = \mathbf{X}\mathbf{X}^\top$$

Representação Matricial de Dados

Exemplo 3 (continuação):

Estimar matriz de covariância, $\boldsymbol{\Sigma}_{\mathbf{y}}$, de $\mathbf{Y}$.

2. Através dum produto matricial:

$$\mathbf{X}\mathbf{X}^{\top} = \begin{bmatrix} x_1[1] & x_1[2] & \cdots & x_1[N] \\ x_2[1] & x_2[2] & & x_2[N] \\ \vdots & & \vdots & \\ x_d[1] & x_d[2] & \cdots & x_d[N] \end{bmatrix} \begin{bmatrix} x_1[1] \cdots x_d[1] \\ x_1[2] \cdots x_d[2] \\ \vdots \\ x_1[N] \cdots x_d[N] \end{bmatrix}$$

$$= \sum_n \begin{bmatrix} x_1^2[n] & x_1[n]x_2[n] & \cdots & x_1[n]x_d[n] \\ x_1[n]x_2[n] & x_2^2[n] & \cdots & x_2[n]x_d[n] \\ \vdots & & \ddots & \\ x_d[n]x_1[n] & x_d[n]x_2[n] & \cdots & x_d^2[n] \end{bmatrix} = \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^{\top}$$

Representação Matricial de Dados

Exemplo 3 (continuação):

Estimar matriz de covariância, $\mathbf{\Sigma_y}$, de $\mathbf{Y}$.

3. Analiticamente:
   - $\mathbf{y} = A\mathbf{x} + \mathbf{b}$
   - $\mathbf{x} \sim \mathcal{N}(\mu_\mathbf{x}, \Sigma_\mathbf{x})$ com $\mu_\mathbf{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ e $\mathbf{\Sigma_x} = \mathbb{E}\left\{\mathbf{x}\mathbf{x}^\top\right\} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}$

   $\mu_\mathbf{y} = \mathbb{E}\left\{\mathbf{y}\right\} = \mathbb{E}\left\{A\mathbf{x} + \mathbf{b}\right\} = A\mathbb{E}\{\mathbf{x}\} + \mathbf{b} = \mathbf{b}$

$$
\begin{aligned}
\mathbf{\Sigma_y} &= \mathbb{E}\left\{(\mathbf{y} - \mu_\mathbf{y})(\mathbf{y} - \mu_\mathbf{y})^\top\right\} = \mathbb{E}\left\{(A\mathbf{x} + \mathbf{b} - \mathbf{b})(A\mathbf{x} + \mathbf{b} - \mathbf{b})^\top\right\} \\
&= \mathbb{E}\left\{A\mathbf{x}\mathbf{x}^\top A^\top\right\} = A\mathbb{E}\left\{\mathbf{x}\mathbf{x}^\top\right\} A^\top = A\mathbf{\Sigma_x}A^\top = AA^\top \\
AA^\top &= \begin{bmatrix} \frac{1}{3\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{3\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{3\sqrt{2}} & \frac{-1}{3\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix}
\end{aligned}
$$

Representação Matriciais de Dados

Exemplo 4: v.a. 2D com covariância pré-definida (`L1AAex004.py`)

1. Gerar 1000 v.a. gaussianas 2D com $\boldsymbol{\mu}_1 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ e $\boldsymbol{\Sigma}_1 = \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix}$
2. Gerar 1000 v.a. uniformes 2D com $\boldsymbol{\mu}_2 = \begin{bmatrix} -3 \\ +2 \end{bmatrix}$ e $\boldsymbol{\Sigma}_2 = \begin{bmatrix} \frac{5}{9} & -\frac{4}{9} \\ -\frac{4}{9} & \frac{5}{9} \end{bmatrix}$

Abordagem:

- Gerar pontos 2D com $\boldsymbol{\mu} = \mathbf{0}$ e $\boldsymbol{\Sigma} = \mathbf{I}$
  - Gerar matriz $\mathbf{X}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ (basta usar `randn()`)
  - Gerar matriz $\mathbf{X}_2$ com `rand()`, para v.a. 1D, $\mu = \frac{1}{2}$ e $\sigma^2 = \frac{1}{12}$
  - $\mathbf{X}_2$ são v.a. iid (matriz de covariância diagonal $\sigma^2\mathbf{I}$).
  - Subtrair $\frac{1}{2}$ a $\mathbf{X}_2$ e dividir por $\frac{1}{\sqrt{12}}$ para $\boldsymbol{\mu} = \mathbf{0}$ e $\boldsymbol{\Sigma} = \mathbf{I}$
- Com $\boldsymbol{\Sigma}_\mathbf{x} = \mathbf{I}$ e $\boldsymbol{\mu}_\mathbf{x} = \mathbf{0}$, e para $\mathbf{y} = A\mathbf{x} + \mathbf{b}$ obtém-se $\boldsymbol{\mu}_\mathbf{y} = \mathbf{b}$ e $\boldsymbol{\Sigma}_\mathbf{y} = AA^\top = A^2$
  - Multiplicar as matrizes de dados $\mathbf{X}_{1,2}$ por $A_{1,2} = \boldsymbol{\Sigma}_{1,2}^{\frac{1}{2}}$
    (para a raiz quadrada matricial, usar `scipy.linalg.sqrtm()`)
  - Adicionar as médias $\boldsymbol{\mu}_{1,2}$

Representação Matriciais de Dados

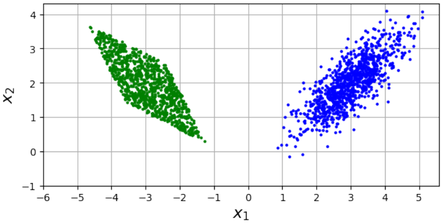
Exemplo 4: v.a. 2D com covariância pré-definida (`L1AAex004.py`)

```
# -*- coding: latin-1 -*-
import numpy as np
import scipy.linalg as la
from matplotlib import pyplot as plt
#cov + média
S1=np.array([[5./9,4./9.],[4./9,5./9]]);m1=np.array([3,2])
S2=np.array([[5./9,-4./9.],[-4./9,5./9]]);m2=np.array([-3,2])
#criar 1000 pontos
N=100000;np.random.seed(0);X1=np.random.randn(2,N)
X2=(np.random.rand(2,N)-0.5)*np.sqrt(12.)#tirar média e por var=1
A1=la.sqrtm(S1);A2=la.sqrtm(S2)
#transformações
Y1=(np.dot(A1,X1).T+m1).T;Y2=(np.dot(A2,X2).T+m2).T
plt.figure(figsize=(7,4)) #criar figura
plt.plot(Y1[0,:],Y1[1,:],'.b',Y2[0,:],Y2[1,:],'.g',markersize=4)
plt.axis('scaled');plt.grid()
plt.xticks(np.arange(-6,6,1));plt.yticks(np.arange(-1,5,1))
plt.xlabel('$x_1$',fontsize=16);plt.ylabel('$x_2$',fontsize=16)
#plt.savefig('../figs/L1AAex004.png',\
#bbox_inches='tight',transparent=True) #guardar em ficheiro ".png"
plt.show()
```

Representação Matriciais de Dados

Exemplo 4: v.a. 2D com covariância pré-definida (`L1AAex004.py`)

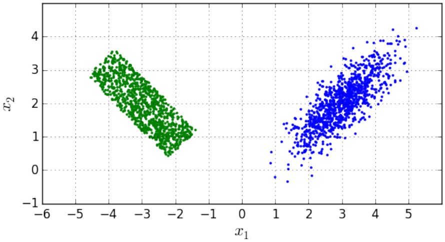
$$A_1 = \sqrt{\begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix}} \qquad A_2 = \sqrt{\begin{bmatrix} \frac{5}{9} & -\frac{4}{9} \\ -\frac{4}{9} & \frac{5}{9} \end{bmatrix}}$$



$$\mathbf{\Sigma}_1 = \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix} \qquad \mathbf{\Sigma}_2 = \begin{bmatrix} \frac{5}{9} & -\frac{4}{9} \\ -\frac{4}{9} & \frac{5}{9} \end{bmatrix}$$

Diferentes transformações podem resultar em dados com as mesmas médias e covariâncias (`L1AAex004b.py`)

$$A_1 = \begin{bmatrix} \frac{1}{3\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{3\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \qquad A_2 = \begin{bmatrix} \frac{1}{3\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{3\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$



$$\mathbf{\Sigma}_1 = \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix} \qquad \mathbf{\Sigma}_2 = \begin{bmatrix} \frac{5}{9} & -\frac{4}{9} \\ -\frac{4}{9} & \frac{5}{9} \end{bmatrix}$$

Transformações Lineares de Dados

Manipular os dados através de transformações lineares é essencial em vários métodos de aprendizagem automática. Transformações lineares surgem igualmente no contexto da geometria projectiva, uma ferramenta indispensável para computação gráfica e para processamento de imagem e visão.

Tranformações Lineares servem para

- Pre-processar dados
- Representar dados mais sucintamente (ex. PCA)
- Visualizar dados com mais que 3 dimensões
- Encontrar padrões nos dados
- Reduzir a quantidade de informação a ser processada
- Separar dados por classe (ex. LDA)
- ...

Transformações Lineares de Dados

Transformação genérica: $\mathbf{y} = A\mathbf{x} + \mathbf{b}$

$$\underset{c \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_c \end{bmatrix}} = \underset{\substack{c \times d \\ \text{transformação linear}}}{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1d} \\ a_{21} & a_{22} & a_{23} & & a_{2d} \\ \vdots & & & \ddots & \\ a_{c1} & a_{c2} & a_{c3} & \cdots & a_{cd} \end{bmatrix}} \underset{d \times 1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix}} + \underset{\substack{c \times 1 \\ \text{translação}}}{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix}}$$

- A adição não é uma operação linear (não cumpre o princípio da sobreposição).
- No entanto, a adição em coordenadas homogéneas é efectuada através de uma multiplicação matricial (operação linear).
- **Notar que:** quando temos um conjunto de N pontos a d dimensões representados numa matrix $\mathbf{X}$ de $d \times N$, a multiplicação matricial, $\mathbf{y} = \mathbf{Ax}$, pode ser aplicada directamente a todo o conjunto: $\mathbf{Y} = \mathbf{AX}$.

Transformações Lineares de Dados

Transformação genérica: $\mathbf{y} = A\mathbf{x} + \mathbf{b}$

$$\underset{c \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_c \end{bmatrix}} = \underset{c \times d}{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1d} \\ a_{21} & a_{22} & a_{23} & & a_{2d} \\ \vdots & & & \ddots & \\ a_{c1} & a_{c2} & a_{c3} & \cdots & a_{cd} \end{bmatrix}} \underset{d \times 1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix}} + \underset{c \times 1}{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix}}$$

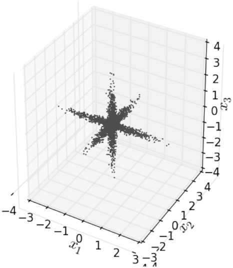
2. Caso $c < d$
   - Dados transformados com menor dimensão que os dados originais
   - Transformação não reversível (há perda de informação)

Transformações Lineares de Dados

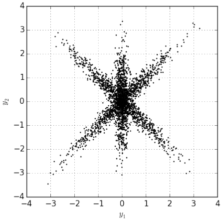
Transformação genérica: $\mathbf{y} = A\mathbf{x} + \mathbf{b}$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_c \end{bmatrix}}_{c \times 1} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1d} \\ a_{21} & a_{22} & a_{23} & & a_{2d} \\ \vdots & & & \ddots & \\ a_{c1} & a_{c2} & a_{c3} & \cdots & a_{cd} \end{bmatrix}}_{c \times d} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix}}_{d \times 1} + \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix}}_{c \times 1}$$

2. Caso $c < d$ - Exemplo: transformação $\mathbb{R}^3 \longrightarrow \mathbb{R}^2$



$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$



Transformações Lineares de Dados

Transformação genérica: $\mathbf{y} = A\mathbf{x} + \mathbf{b}$

$$\underset{c \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_c \end{bmatrix}} = \underset{c \times d}{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1d} \\ a_{21} & a_{22} & a_{23} & & a_{2d} \\ \vdots & & & \ddots & \\ a_{c1} & a_{c2} & a_{c3} & \cdots & a_{cd} \end{bmatrix}} \underset{d \times 1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix}} + \underset{c \times 1}{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix}}$$

2. Caso $c > d$
   - Dados transformados com maior dimensão que os dados originais
   - Transformação pode ser reversível[2]
   - Quando é reversível, não há perda nem ganho de informação
   - Dados continuam num sub-espaço com a mesma dimensão dos dados originais

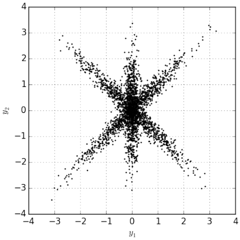
[2] No entanto, se a matriz conter linhas ou colunas linearmente dependentes uma das outras, pode não o ser. Exemplo: uma matriz só com 0s não é obviamente reversível.

Transformações Lineares de Dados

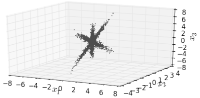
Transformação genérica: $\mathbf{y} = A\mathbf{x} + \mathbf{b}$

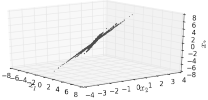
$$
\underset{c \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_c \end{bmatrix}} = \underset{c \times d}{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1d} \\ a_{21} & a_{22} & a_{23} & & a_{2d} \\ \vdots & & & \ddots & \\ a_{c1} & a_{c2} & a_{c3} & \cdots & a_{cd} \end{bmatrix}} \underset{d \times 1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix}} + \underset{c \times 1}{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix}}
$$

2. Caso $c > d$ - Exemplo: transformação $\mathbb{R}^2 \longrightarrow \mathbb{R}^3$



$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
$$





(outra perspectiva)

Transformações Lineares de Dados

Transformação genérica: $\mathbf{y} = A\mathbf{x} + \mathbf{b}$

$$\underset{c \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_c \end{bmatrix}} = \underset{c \times d}{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1d} \\ a_{21} & a_{22} & a_{23} & & a_{2d} \\ \vdots & & & \ddots & \\ a_{c1} & a_{c2} & a_{c3} & \cdots & a_{cd} \end{bmatrix}} \underset{d \times 1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix}} + \underset{c \times 1}{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix}}$$

3. Caso $c = d$
   Matrizes Quadradas - transformação $\mathbb{R}^d \longrightarrow \mathbb{R}^d$
   - Dados transformados com a mesma dimensão que os dados originais
   - Transformação reversível quando $\det(A) \neq 0$ (determinante $\neq 0$)
   - Se $\det(A) \neq 0$, então $\mathbf{x} = A^{-1}(\mathbf{y} - \mathbf{b})$

Transformações Lineares de Dados

Matrizes Quadradas

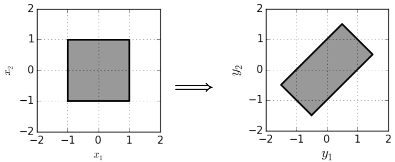
Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Determinantes:**
  O valor absoluto de $\det(A)$ corresponde ao escalamento por unidade de área depois da transformação.
  Exemplo:

$$\mathbf{y} = A\mathbf{x} = \begin{bmatrix} 1.0 & -0.5 \\ 1.0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



$$|\det(A)| = 1.0 \quad \Longrightarrow \quad \text{quadrado e rectânglo têm a mesma área}$$

Transformações Lineares de Dados

Matrizes Quadradas

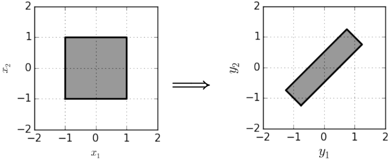
Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Determinantes:**
  O valor absoluto de $\det(A)$ corresponde ao escalamento por unidade de área depois da transformação.
  Exemplo:

$$\mathbf{y} = A\mathbf{x} = \begin{bmatrix} 1.0 & -0.25 \\ 1.0 & 0.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



$|\det(A)| = 0.5 \implies$ área do rectângulo é metade da área do quadrado

Transformações Lineares de Dados

Matrizes Quadradas

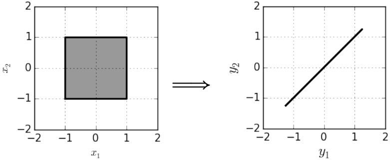
Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Determinantes:**
  O valor absoluto de $\det(A)$ corresponde ao escalamento por unidade de área depois da transformação.
  Exemplo:

$$\mathbf{y} = A\mathbf{x} = \begin{bmatrix} 1.0 & 0.25 \\ 1.0 & 0.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



$|\det(A)| = 0 \implies$ área de retângulo é 0. (espaço 2D “achatado” a 1D)

Transformações Lineares de Dados

Matrizes Quadradas

Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Escalamentos, Rotações e Translações:**
  Em aprendizagem automática, estas transformações servem principalmente para representar os dados de uma forma mais informativa ou para que certos padrões ou características possam sobressair. Estas operações são igualmente utilizadas noutros contextos, nomeadamente em computação gráfica e processamento de imagem e visão. Escalamentos, rotações e translações são **transformações afins**. Escalamentos e rotações são feitos através de um produto matricial e translações através da soma de um vector.

Escalamento $\mathbf{y} = \mathbf{\Delta x}$ onde $\mathbf{\Delta}$ é uma matriz diagonal $d \times d$ (d -dimensão dos dados)

Rotação $\mathbf{y} = \mathbf{Ux}$ com $\mathbf{U}$ sendo uma matriz de rotação (e ortogonal) $d \times d$

Translação $\mathbf{y} = \mathbf{x} + \mathbf{b}$ com $\mathbf{b}$ vetor $d \times 1$

Reflexão $\mathbf{y} = -\mathbf{x}$ (reflexão na origem)

Transformações Lineares de Dados

Matrizes Quadradas

Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Escalamentos:**

$$\mathbf{y} = \mathbf{\Delta x} = \begin{bmatrix} \delta_1 & \cdots & 0 \\ \vdots & \ddots & \\ 0 & \cdots & \delta_d \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} \delta_1 x_1 \\ \vdots \\ \delta_d x_d \end{bmatrix}$$

Um escalamento corresponde a multiplicar **x** por uma matriz diagonal. Cada dimensão, x_i, do vector **x** é escalada por um factor δ_i.

Transformações Lineares de Dados

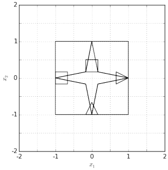
Matrizes Quadradas

Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

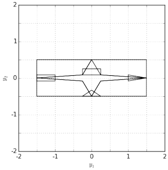
$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Escalamentos:**
  Exemplo:



$$\mathbf{y} = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}$$

$$\longrightarrow$$



Transformações Lineares de Dados

Matrizes Quadradas

Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

$$\underset{2 \times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2 \times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2 \times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2 \times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Rotações:**

$$\mathbf{y} = \mathbf{U}\mathbf{x} = \begin{bmatrix} \cos(\theta) & \mp\sin(\theta) \\ \pm\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1\cos(\theta) \mp x_2\sin(\theta) \\ \pm x_1\sin(\theta) + x_2\cos(\theta) \end{bmatrix}$$

onde θ é um ângulo.

$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ +\sin(\theta) & \cos(\theta) \end{bmatrix}$ rotação no sentido contrário aos ponteiros do relógio

$\begin{bmatrix} \cos(\theta) & +\sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$ rotação no sentido dos ponteiros do relógio

Transformações Lineares de Dados

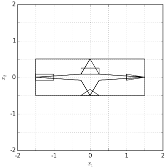
Matrizes Quadradas

Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

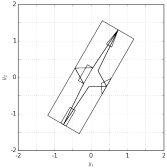
$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Rotações:**
  Exemplo:



$$\mathbf{y} = \begin{bmatrix} \cos(\frac{\pi}{3}) & -\sin(\frac{\pi}{3}) \\ \sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) \end{bmatrix} \mathbf{x}$$

$$\longrightarrow$$



Transformações Lineares de Dados

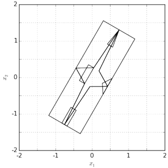
Matrizes Quadradas

Caso de estudo: espaços 2D

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

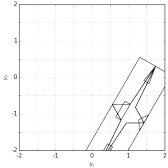
$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- **Translações:**
  Exemplo:



$$\mathbf{y} = \mathbf{x} + \begin{bmatrix} +1 \\ -1 \end{bmatrix}$$

$$\longrightarrow$$



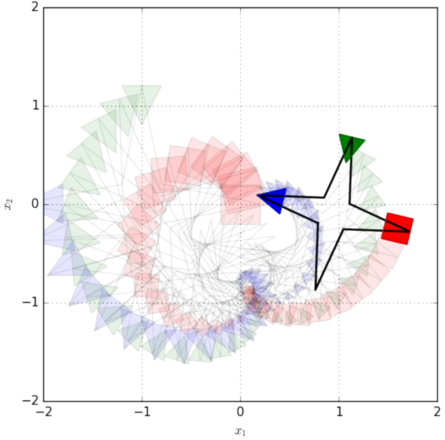
Transformações Lineares de Dados

Escalamentos, Rotações, Translações (L1AAex006.py)

```
# -*- coding: latin-1 -*-
import numpy as np
from matplotlib import pyplot as plt
import pickle
Parm=pickle.load(open('L1AAestrela.p','rb'))#importar pontos
pts0=Parm[0];pts1=Parm[1];pts2=Parm[2];pts3=Parm[3]
N=40 #nº de transformações
ang=np.arange(-np.pi,2.*np.pi,3.*np.pi/N)
transl=np.zeros((2,N))
transl[0,:]=np.arange(-1,1,2./N);transl[1,:]=transl[0,:]**2-1
scl=(1.+np.cos(transl[0,:]*np.pi/2+np.pi/2))-.8#escalamento
plt.figure(figsize=(7,6.5))     #criar figura
idx=0;media=np.zeros((2,1))     #vector temporario
for a in ang:
    T=np.array([[np.cos(a),-np.sin(a)],[np.sin(a),np.cos(a)]])
    media[0,0]=transl[0,idx];media[1,0]=transl[1,idx]
    s=scl[idx]
    idx=idx+1
    x0=np.dot(T,s*pts0)+media;x1=np.dot(T,s*pts1)+media
    x2=np.dot(T,s*pts2)+media;x3=np.dot(T,s*pts3)+media
    plt.plot(x0[0,:],x0[1,:],'-k',alpha=.1)
    plt.fill(x1[0,:],x1[1,:],'b',x2[0,:],x2[1,:],'g',\
    x3[0,:],x3[1,:],'r',alpha=.1)
```

Transformações Lineares de Dados

Escalamentos, Rotações, Translações (`L1AAex006.py`)



Transformações Lineares de Dados

Coordenadas Homogéneas:

Coordenadas homogéneas são usadas em geometria projectiva tal como as coordenadas cartesianas são usadas em geometria Euclideana. Formulas expressas em coordenadas homogéneas são geralmente mais simples do que a formulação em coordenadas cartesianas, e por isso, são muito usadas computação gráfica, onde transformações afins e projectivas são representadas com uma só matriz.

Caso de estudo: espaços 2D

- Coordenadas cartesianas

$$\mathbf{y} = A\mathbf{x} + \mathbf{b}$$

$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 2}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{2\times 1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

- Coordenadas homogéneas

$$\mathbf{y} = A\mathbf{x} + \mathbf{b} = M\mathbf{x}'$$

$$\underset{2\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \underset{2\times 3}{\begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{bmatrix}} \underset{3\times 1}{\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}}$$