Notas de Aula de Computação Científica e Análise de Dados (DRAFT)

João Victor Lopez Pereira
Prof. Dr. João Antonio Recio da Paixão

27 de novembro de 2024

Rio de Janeiro - RJ

Sumário

Informações Gerais

Esse documento contém notas de aula da disciplina *Computação Científica e Análise de Dados*, oferecida na Universidade Federal do Rio de Janeiro (UFRJ). O objetivo do documento é fornecer um material de apoio para os estudantes da disciplina e interessados na área.

As notas podem conter erros ou estar desatualizadas. Caso encontre alguma imprecisão, incoerência ou tenha sugestões, sinta-se à vontade para entrar em contato comigo pelo *e-mail* joaovlp@dcc.ufrj.br. Assim, poderei corrigir rapidamente o problema e disponibilizar uma versão aprimorada para todos.

Atualmente, o documento contém diversas *tags TODO*, que funcionam como marcadores para futuras melhorias (como "inserir uma imagem sobre este tópico"). Além disso, várias imagens genéricas estão presentes como *placeholders* e serão, em breve, substituídas por ilustrações mais apropriadas para facilitar a compreensão dos assuntos abordados. O capítulo 2, por sua vez, está em branco pelo mesmo motivo e será atualizado em breve.

Capítulo 1

Ajuste de Curvas e Problema dos Mínimos Quadrados

1.1 Motivação para Mínimos Quadrados

Dado uma lista de pontos $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ sendo n seu tamanho, queremos uma função f polinomial tal que $\forall k \in \{1, \ldots, n\}$, $f(x_k) = y_k$. Ou seja, uma função que passe por todos os pontos dados.

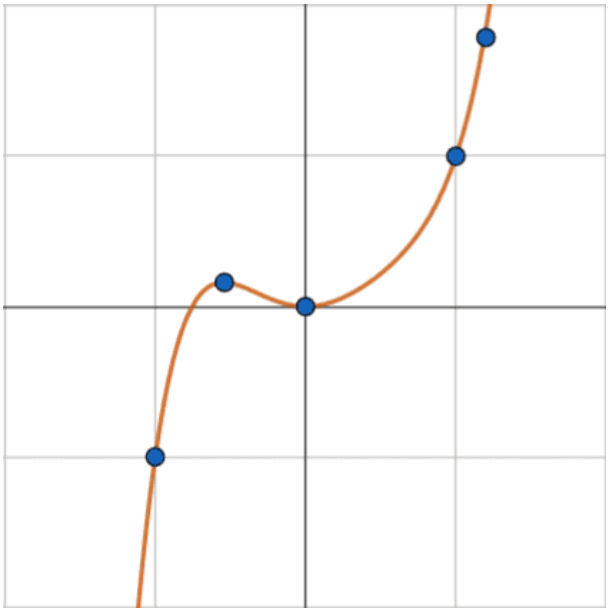

Figura 1.1: Função passando por pontos aleatórios.

Veja que, se os pontos fornecidos não estiverem em formato de reta (polinômio de 1° ou 0° grau), precisaremos necessariamente que f seja uma função de grau maior que 1 para que f passe por todos os pontos fornecidos.

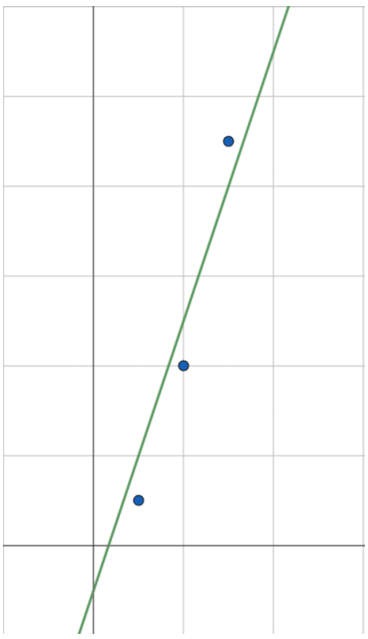

Figura 1.2: Aproximação via função de 1° grau.

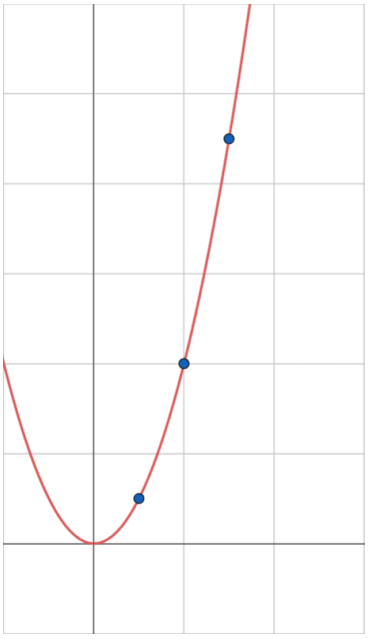

Figura 1.3: Aproximação via função de 2° grau.

O problema é que, quando muitos pontos são dados, a função resultante do processo de interpolação está sujeita a ter um grau alto, fazendo com que a função f de saída esteja sujeita a diferentes "penalidades", como o fenômeno de Runge.

1.2 Solução para Regressões Polinomiais

Sendo assim, gostaríamos de uma função f de grau m tal que $\forall k \in \{1, \ldots, n\}$, $f(x_k) = y_k$ tal que o grau da função resultante não seja desnecessariamente grande.

$$f(x_k) = y_k \tag{1}$$

Sabendo que f é uma função de grau m, temos:

$$f(x_k) = c_1 + c_2 x_k^1 + \cdots + c_{m+1} x_k^m \tag{2}$$

Substituindo (1) em (2):

$$y_k = c_1 + c_2 x_k^1 + \cdots + c_{m+1} x_k^m \tag{3}$$

Mas veja que, ao utilizar um polinômio de grau m e termos n pontos, existe a possibilidade de termos mais pontos do que o grau do polinômio pode cobrir. Por exemplo, não é possível

que uma função no formato $g(x) = ax + b$ respeite $g(1) = 1$, $g(2) = 4$ e $g(3) = 9$ (ou seja, passe pelos pontos $(1, 1)$, $(2, 4)$ e $(3, 9)$).

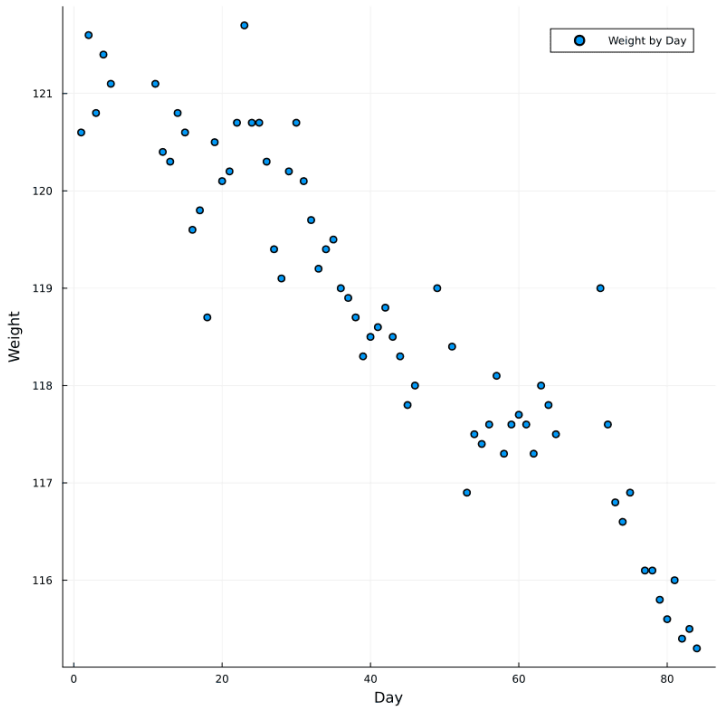

Figura 1.4: Exemplo de situação em que o uso de um polinômio de grau alto não parece útil.

Veja que, no exemplo acima, em que são dados 69 pontos, não parece ser uma boa ideia aproximá-los utilizando um polinômio de grau 68 visto que a função f sofrerá:

1. *Overfitting*: Utilizar um polinômio de grau 68 fará com que, de fato, $f(x_k) = y_k \; \forall k \in \{1, \ldots, n\}$, mas com o custo de bastante ruído por erros de pontos flutuante e péssimo poder de aproximação para novos dados.
2. Complexidade do Modelo: Um polinômio de grau 68 apresenta complexidade maior do que uma simples reta ou função quadrática, além de instabilidade, pior performance computacional e menor capacidade de interpretação.

Veja uma possível aproximação dos dados mostrados no gráfico acima ao utilizarmos um polinômio de 1° grau:

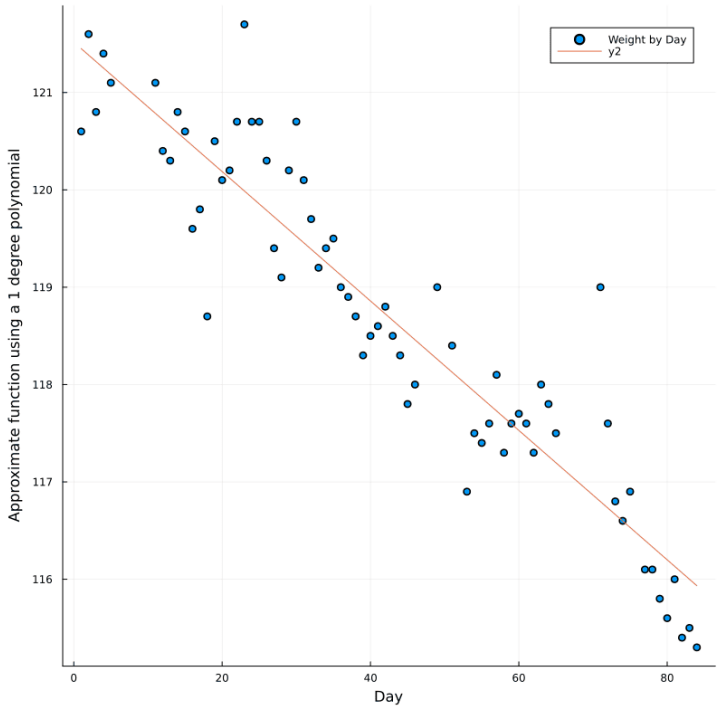

Figura 1.5: Aproximação utilizando polinômio de 1° grau.

Além disso, a escolha a respeito do grau do polinômio utilizado para realizar a aproximação não é necessariamente objetiva. Por exemplo, dado um conjunto de pontos, podemos realizar uma aproximação a partir de um polinômio de 1° grau:

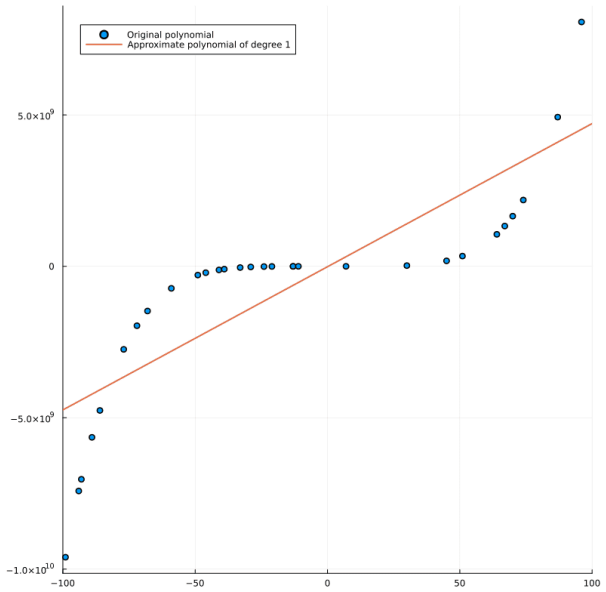

Figura 1.6: Aproximação de pontos utilizando um polinômio de 1° grau.

Mas também poderíamos utilizar um polinômio de grau 3 ou 5:

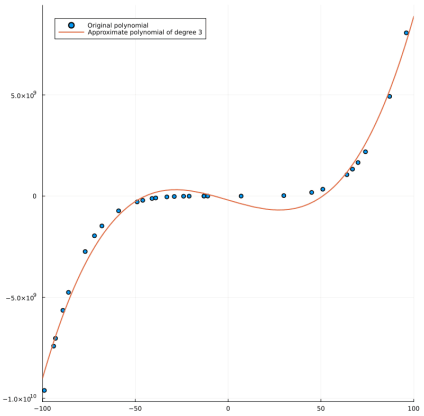

Figura 1.7: Aproximação para os mesmos pontos utilizando um polinômio de 3° grau.

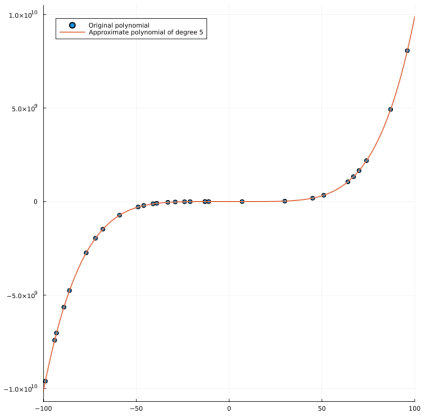

Figura 1.8: Aproximação para os mesmos pontos utilizando um polinômio de 5° grau.

Sendo assim, dado uma lista de pontos $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ gostaríamos de encontrar a melhor função f possível tal que $\forall k \in \{1, \ldots, n\}$, $f(x_k) \approx y_k$. Com isso, teremos um sistema de equações semelhante a (3):

$$y_k \approx c_1 + c_2 x_k^1 + \cdots + c_{m+1} x_k^m$$

Sabemos que isso é verdade $\forall k \in \{1, \ldots, n\}$, logo, $\forall i \in \{1, \ldots, n\}$:

$$\begin{aligned}
y_1 &\approx c_1 + c_2 x_1^1 + \cdots + c_{m+1} x_1^m \\
&\vdots \\
y_i &\approx c_1 + c_2 x_i^1 + \cdots + c_{m+1} x_i^m \\
&\vdots \\
y_n &\approx c_1 + c_2 x_n^1 + \cdots + c_{m+1} x_n^m
\end{aligned}$$

Sendo assim, podemos escrever essas equações em um sistema matriz-vetor:

$$\underbrace{\begin{bmatrix} 1 & \ldots & x_1^{j-1} & \ldots & x_1^m \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & \ldots & x_i^{j-1} & \ldots & x_i^m \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & \ldots & x_n^{j-1} & \ldots & x_n^m \end{bmatrix}}_{\mathcal{A}_{(n \times m+1)}} \underbrace{\begin{bmatrix} c_1 \\ \vdots \\ c_j \\ \vdots \\ c_{m+1} \end{bmatrix}}_{cs_{(m+1 \times 1)}} \approx \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix}}_{ys_{(n \times 1)}}$$

tal que $i \in \{1, \ldots, n\}$ e $j \in \{1, \ldots, m+1\}$.

Mas veja que esse sistema não será exato para todo valor m escolhido (grau do polinômio utilizado para aproximar os pontos), logo, não podemos resolver esse sistema por meio da Eliminação Gaussiana e da Substituição Reversa.

Também podemos representar esse sistema matriz-vetor na forma

$$c_1 \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \cdots + c_j \begin{bmatrix} x_1^{j-1} \\ \vdots \\ x_i^{j-1} \\ \vdots \\ x_n^{j-1} \end{bmatrix} + \cdots + c_{m+1} \begin{bmatrix} x_1^m \\ \vdots \\ x_i^m \\ \vdots \\ x_n^m \end{bmatrix} \approx \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix}$$

e podemos ver esse sistema como um subespaço gerado pelos vetores da matriz $\mathcal{A}$ e os coeficientes sendo o quanto daqueles vetores o vetor dos ys precisa para ser representado. Podemos visualizar esse sistema no caso de uma matriz com 2 colunas (ou seja, $m = 1$):

Todo: Melhorar esse desenho (Trazer características que o faça parecer estar no $\mathbb{R}^3$).

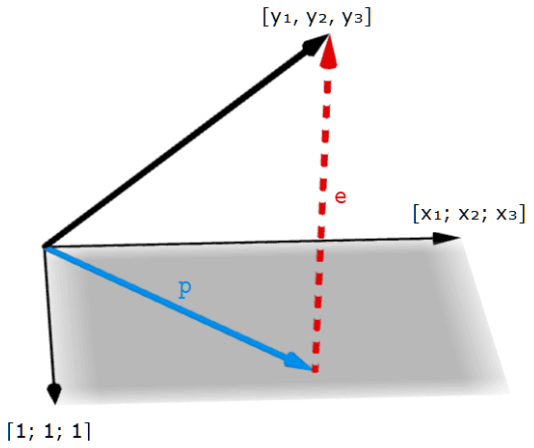

Figura 1.9: Visualização do problema.

Veja que o vetor dos ys não necessariamente estará no plano gerado pelos vetores da matriz $\mathcal{A}$ (nesse caso $[1; 1; 1]$ e $[x_1; x_2; x_3]$). Veja que a melhor aproximação possível para ys será a sua projeção no plano visto que a distância entre ys e sua projeção é a menor distância possível do vetor ao plano. Sendo assim:

$$E(c_1, c_2) = \left\| c_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \right\|^2$$

Ou seja, no caso geral, o erro E do sistema pode ser calculado como:

$$E(cs) = \|\mathcal{A}cs - ys\|^2$$

No caso anterior:

$$\begin{aligned} E(c_1, c_2) &= \left\| c_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} c_1 + c_2 x_1 - y_1 \\ c_1 + c_2 x_2 - y_2 \\ c_1 + c_2 x_3 - y_3 \end{bmatrix} \right\|^2 \\ &= \sqrt{(c_1 + c_2 x_1 - y_1)^2 + (c_1 + c_2 x_2 - y_2)^2 + (c_1 + c_2 x_3 - y_3)^2}^2 \\ &= (c_1 + c_2 x_1 - y_1)^2 + (c_1 + c_2 x_2 - y_2)^2 + (c_1 + c_2 x_3 - y_3)^2 \end{aligned}$$

Perceba que essa equação forma um paraboloide:

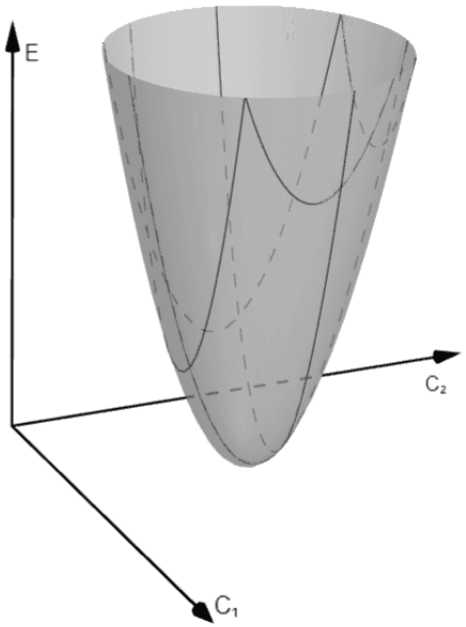

Figura 1.10: Paraboloide semelhante ao formado pela função $E(c_1, c_2)$.

Sendo assim, queremos c_1 e c_2 tais que $E(c_1, c_2)$ seja o menor possível. Em termos da figura 1.10, podemos dizer que estamos buscando c_1 e c_2 tais que $\nabla E(c_1, c_2) = 0$ (ponto mínimo da função do erro).

Visto que a melhor aproximação possível para ys é sua projeção no plano gerado por $\mathcal{A}$, ao invés de fazer $c_1 + c_2 xs^1 \approx ys$, faremos $c_1 + c_2 xs^1 = p$ tal que p seja a projeção de ys no plano gerado por $\mathcal{A}$.

Visto também que o erro mínimo ocorre quando E é a distância de ys a p, o vetor e é perpendicular ao plano.

$$e \perp \text{ plano gerado por } \mathcal{A} \implies \mathcal{A}^T e = 0$$

visto que $e = \mathcal{A}cs - ys$:

$$\begin{aligned} \mathcal{A}^T(\mathcal{A}cs - ys) &= 0 \\ \mathcal{A}^T\mathcal{A}cs - \mathcal{A}^T ys &= 0 \\ \mathcal{A}^T\mathcal{A}cs &= \mathcal{A}^T ys \end{aligned}$$

Ou seja, mesmo que $\mathcal{A}cs \approx ys$, temos que $\mathcal{A}^T\mathcal{A}cs = \mathcal{A}^T ys$.

1.3 Solução para Regressão Exponencial e Logarítmica

Infelizmente — para o caso de querermos utilizar uma função em formato exponencial ou logarítmico — não existe um jeito simples e genérico que trate de todos os casos possíveis de funções. No caso polinomial, precisávamos só inserir mais uma coluna na matriz de Vandermonde e mais um coeficiente para aumentar o grau do polinômio, mas no caso de funções como:

$$\begin{aligned} f(x) &= e^{c_2 x} \\ g(x) &= c_1 e^{c_2 x} \\ h(x) &= c_1 e^{c_2 x} + c_3 \end{aligned}$$

Não existe uma forma genérica de tratar. Sendo assim, veremos como: dado xs, ys e um formato para a função f, como encontrar coeficientes que aproximem os dados pontos.

Veja que, dessa vez, estamos tratando de um problema não linear. Sendo assim, precisamos manipular nossa função f para que possamos expressar nossos dados na forma matriz-vetor, resolver o sistema linear e encontrar os coeficientes.

Escolheremos uma função genérica:

Seja

$$f(x) = c_1 e^{c_2 x} + c \tag{1}$$

Dados xs, ys e c constante, Queremos determinar c_1 e c_2 tais que:

$$f(x_k) \approx y_k \quad \forall k \in \{1, \ldots, n\} \tag{2}$$

sendo n a quantidade de pontos. Sendo assim, substituindo (1) em (2):

$$y_k \approx c_1 e^{c_2 x_k} + b$$

Podemos manipular essa equação tal que:

$$\begin{aligned} y_k &\approx c_1 e^{c_2 x_k} + c \\ y_k - c &\approx c_1 e^{c_2 x_k} \\ \ln(y_k - c) &\approx \ln(c_1 e^{c_2 x_k}) \\ \ln(y_k - c) &\approx \ln(c_1) + ln(e^{c_2 x_k}) \\ \ln(y_k - c) &\approx \ln(c_1) + c_2 x_k \end{aligned}$$

Visto que sabemos os valores de y, x e c, o lado esquerdo da equação é um número, enquanto o lado direito contém nossas incógnitas. Seja:

$$\begin{aligned} \bar{ys} &= \ln(y_k - c) \\ \bar{c_1} &= \ln(c_1) \\ \bar{c_2} &= c_2 \\ \bar{xs} &= x_i \ \forall i \in \{1, \ldots, n\} \end{aligned}$$

Perceba que agora temos sistemas na forma:

$$\bar{ys} \approx \bar{c_1} + \bar{c_2}\bar{xs}$$

Perceba que esse sistema está em um formato que já sabemos resolver (pois aparenta ser linear), logo, podemos montá-lo na forma matriz vetor:

$$\begin{bmatrix} \bar{y_1} \\ \vdots \\ \bar{y_i} \\ \vdots \\ \bar{y_n} \end{bmatrix} \approx \begin{bmatrix} 1 & \bar{x_1} \\ \vdots & \vdots \\ 1 & \bar{x_i} \\ \vdots & \vdots \\ 1 & \bar{x_n} \end{bmatrix} \begin{bmatrix} \bar{c_1} \\ \bar{c_2} \end{bmatrix}$$

tal que $i \in \{1, \ldots, n\}$.

Sabemos resolver esse sistema pois é o mesmo que encontramos na seção 1.2.

$$\bar{\mathcal{A}}\bar{cs} \approx \bar{y}\bar{s}$$
$$\bar{\mathcal{A}}^T\bar{\mathcal{A}}\bar{cs} = \bar{\mathcal{A}}^T\bar{y}\bar{s}$$

Após resolver esse sistema, o valor de $\bar{cs}$ encontrado não corresponderá ao valor original, visto que $\bar{c_1} = ln(c_1)$.

Para achar cs:

$$c_1 = e^{\bar{c_1}}$$
$$c_2 = \bar{c_2}$$

Dessa forma, encontramos cs que aproximam o sistema. Infelizmente, não temos a garantia de que os cs encontrados sejam os que aproximam o sistema o melhor possível. Além disso, infelizmente também não há uma formato genérico de sistema que resolva para todas os possíveis formatos de equações, mas o objetivo é — por meio de manipulações algébricas — transformar o sistema não linear em linear.

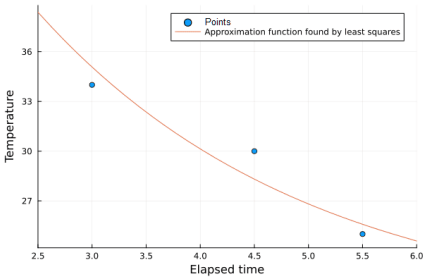

Figura 1.11: Exemplo de aproximação para dados pontos com uma função exponencial.

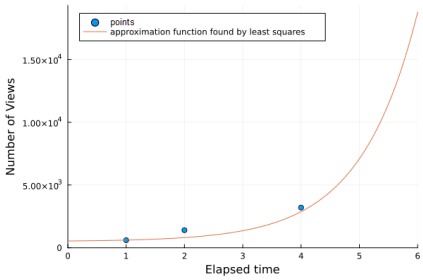

Figura 1.12: Outro exemplo de aproximação para dados pontos com uma função exponencial.

1.4 Matrizes Ortogonais em Mínimos Quadrados

Veja que na seção 1.2 a solução encontrada para que o sistema inexato $Ax \approx b$ se tornasse exato foi multiplicar ambos os lados do sistema pela transposta de A tal que:

$$Ax \approx b$$
$$A^TAx = A^Tb$$

Mas veja que a ordem de complexidade desse algoritmo é $O(n^3)$ por conta do produto entre as matrizes. Sendo assim, gostaríamos de encontrar um método que apresentasse complexidade

menor para encontrarmos a melhor solução possível para um dado sistema $Ax \approx b$. Sendo assim, para que exista uma matriz M tal que $\|Ax - b\|^2 = \|M(Ax - b)\|^2 = \|MAx - Mb\|^2$ e que $(MAx - Mb)$ apresente complexidade menor do que $O(n^3)$ precisamos necessariamente não ter que lidar com o produto entre matrizes (em outras palavras, precisamos não ter que computar o produto MA).

Seja Q uma matriz ortonormal. Sabemos que matrizes ortonormais respeitam $Q^T = Q^{-1}$, logo, respeitam $Q^TQ = I$. Além disso:

$$\begin{aligned}
\|QA\|^2 &= \left\|(QA)^TQA\right\|^2 \\
&= \left\|A^TQ^TQA\right\|^2 \\
&= \left\|A^TA\right\|^2 \\
&= \|A\|^2
\end{aligned}$$

Ou seja, o fato da matriz Q ser ortonormal implica que ela não afetará o valor da norma. Logo, M pode ser ortonormal.

Queremos matrizes Q_i^T tais que:

$$\begin{aligned}
Q_n^T \dots Q_3^TQ_2^TQ_1^TA &= R \\
Q^TA &= R \\
QQ^TA &= QR \\
A &= QR \\
A &= QR
\end{aligned}$$

sendo $Q_1^T, Q_2^T, Q_3^T, \dots, Q_n^T$ e Q matrizes ortonormais tais que $Q = Q_1^T, Q_2^T, Q_3^T, \dots, Q_n^T$, A sendo nossa matriz original e R triangular superior pois, assim:

$$\begin{aligned}
\|Ax - b\|^2 &= \|QRx - b\|^2 \\
&= \left\|Q^T(QRx - b)\right\|^2 \\
&= \left\|Q(Q^T(QRx - b))\right\|^2 \\
&= \left\|Q(Q^TQRx - Q^Tb)\right\|^2 \\
&= \left\|Q(Rx - Q^Tb)\right\|^2 \\
&= \left\|(Rx - Q^Tb)\right\|^2 \\
&= \|(Rx - c)\|^2
\end{aligned}$$

tal que $c = Q^Tb$. Ao sabemos que $Ax \approx b$, sabemos que:

$$\begin{aligned} Ax \approx b &\iff Q^T Ax \approx Q^T b \\ &\iff Rx \approx c \end{aligned}$$

Sendo assim, ao saber que R é uma matriz triangular superior, a ordem de complexidade para encontrar x tal que $Rx = c$ é $O(n^2)$.

Por exemplo, ao termos um sistema de equação $Ax = b$:

$$\begin{bmatrix} 12 & 24 \\ 4 & 14 \\ -3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -62 \\ 104 \\ -53 \end{bmatrix}$$

Podemos escrevê-lo na forma $Rx = c$ tal que:

$$[A \,|\, b] = Q \,[R \,|\, c]$$

No caso de nossa A e b:

$$\begin{bmatrix} 12 & 24 & | & -62 \\ 4 & 14 & | & 104 \\ -3 & 2 & | & -53 \end{bmatrix} = Q \begin{bmatrix} -13 & -26 & | & 13 \\ 0 & -10 & | & -20 \\ 0 & 0 & | & -130 \end{bmatrix}$$

Sendo assim, na fórmula do erro temos:

$$\begin{aligned} E &= \|Ax - b\|^2 \\ &= \|Rx - c\|^2 \\ &= \left\| \begin{bmatrix} -13 & -26 \\ 0 & -10 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 13 \\ -20 \\ -130 \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} -13x_1 & + & -26x_2 \\ 0 & + & -10x_2 \\ 0 & + & 0 \end{bmatrix} - \begin{bmatrix} 13 \\ -20 \\ -130 \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} -13x_1 & + & -26x_2 & + & -13 \\ 0 & + & -10x_2 & + & 20 \\ 0 & + & 0 & + & -130 \end{bmatrix} \right\|^2 \\ &= \sqrt{(-13x_1 + -26x_2 + -13)^2 + (0 + -10x_2 + 20)^2 + (0 + 0 + 130)^2}^2 \\ &= (-13x_1 + -26x_2 + -13)^2 + (0 + -10x_2 + 20)^2 + (0 + 0 + 130)^2 \\ &= (-13x_1 + -26x_2 + -13)^2 + (-10x_2 + 20)^2 + (130)^2 \end{aligned}$$

Veja que podemos escolher os valores $x_2 = \dfrac{20}{10} = 2$, temos:

$$\begin{aligned} E &= (-13x_1 + -26x_2 + -13)^2 + (-10x_2 + 20)^2 + (130)^2 \\ &= (-13x_1 - 65)^2 + 0 + (130)^2 \end{aligned}$$

E ao escolher $x_1 = \dfrac{-65}{13} = -5$, temos:

$$\begin{aligned} E &= (-13x_1 - 65)^2 + 0 + (130)^2 \\ &= 0 + 0 + (130)^2 \\ &= 130^2 \end{aligned}$$

Sendo assim, pode-se perceber que o erro depende somente da parcela do vetor c ao lado da parcela de R em que R não apresenta variáveis visto que indiferente dos valores de x_i que escolhermos, essa parcela continuará necessariamente ocasionando em $0 = c_2$.

$$\begin{aligned} \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix} x &= \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \\ \bar{R}x &= c_1 \\ 0x &= c_2 \end{aligned}$$

Sendo assim, podemos concluir que o erro do método pode ser calculado como:

$$\begin{aligned} E &= \|Ax - b\|^2 \\ &= \left\|Q^T Ax - Q^T b\right\|^2 \\ &= \left\| \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix} x - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} \bar{R}x - c_1 \\ 0 - c_2 \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} \bar{R}x - c_1 \\ -c_2 \end{bmatrix} \right\|^2 \\ &= \left\|\bar{R}x - c_1\right\|^2 + \|-c_2\|^2 \\ &= 0 + \|-c_2\|^2 \\ &= \|c_2\|^2 \end{aligned}$$

Capítulo 2

Decomposição QR

2.1 Motivação para Decomposição QR

2.2 Algoritmo de Reflexão de Householder para Decomposição QR

2.3 Algoritmo de Rotação de Givens para Decomposição QR

2.4 Algoritmo de Gram-Schmidt para Decomposição QR

Capítulo 3

Transformadas

3.1 Motivação para Transformadas

Suponha que as notas musicais possam ser representadas simplesmente por funções senos ou cossenos. Dado duas notas n_1 e n_2 com frequências diferentes, podemos simplesmente juntá-las em uma combinação linear para gerar uma outra onda o tal que:

$$o = c_1\, n_1 + c_2\, n_2$$

E c_1 e c_2 sejam as intensidades que cada nota foi tocada.

Suponha que n_1 se refira à nota C na oitava 0 e n_2 se refira a nota E na oitava 1. Sendo assim, temos notas que, respectivamente, apresentam frequências aproximadamente $16Hz$ e $41Hz$. Representaremos essas notas utilizando a letra a qual ela se refere junto a sua oitava. Nesse caso, a primeira nota será referida como $C0$ e a segunda nota será referida como $E1$. Podemos visualizar uma representação para essas notas nos gráficos abaixo.

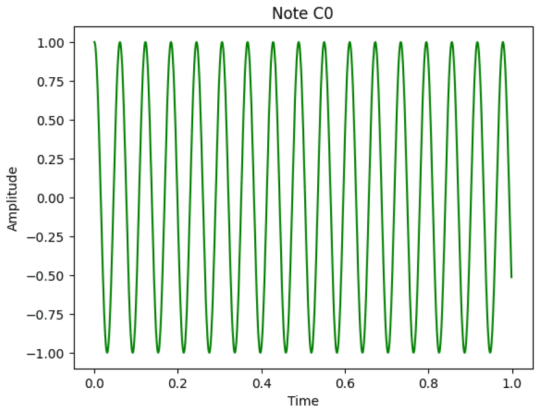

Figura 3.1: Representação da nota $C0$.

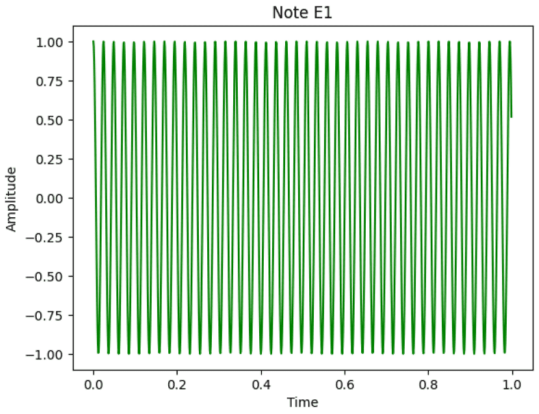

Figura 3.2: Representação da nota $E1$.

Se juntarmos essas notas com intensidades (coeficientes) 1.2 e 0.4, respectivamente, temos a

equação:

$$o = 1.2\,C0 + 0.4\,E1$$

Perceba que, como c_0 e c_1 são funções senos ou cossenos, o também será.

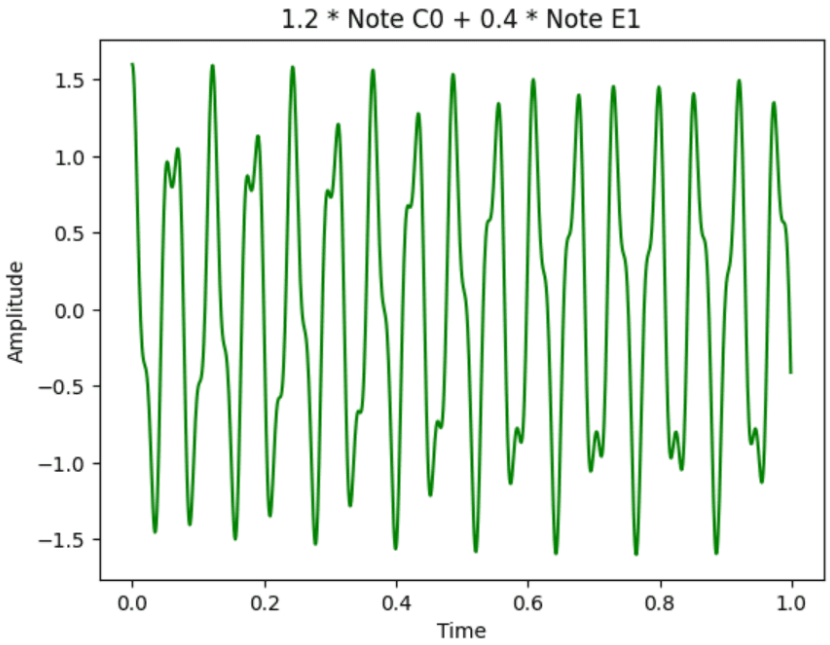

Figura 3.3: Representação da onda gerada pela combinação linear de n_1 e n_2.

Juntar as notas n_1 e n_2 não é um problema. A motivação por trás das transformadas vistas nesse capítulo são: dado uma onda o, como separá-las nas ondas n_1 e n_2?

3.2 Transformada Rápida de Walsh-Hadamard

Quando tínhamos pontos xs e ys no plano cartesiano, podíamos simplesmente fazer realizar uma interpolação para encontrar uma função que passe por esses pontos. Por exemplo, em uma interpolação cúbica:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Que é o mesmo que:

$$c_1 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + c_3 \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \\ x_4^2 \end{bmatrix} + c_4 \begin{bmatrix} x_1^3 \\ x_2^3 \\ x_3^3 \\ x_4^3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Veja que, na segunda representação do sistema, estamos tentando determinar um coeficiente que multiplica um componente que contém xs de determinadas potências. Em outras palavras, perceba que esse coeficiente determina o quanto daquele vetor é necessário para expressar os ys, mas perceba que cada vetor representa uma grau de polinômio:

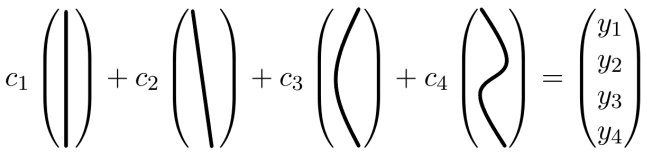

Figura 3.4: Perspectiva diferente da Matriz de Vandermonde.

Faremos algo semelhante para conseguir separar a onda o em dois componentes, com o porém de utilizarmos uma representação para funções senos e cossenos ao invés de monômios em cada coluna da matriz. Sendo assim, temos um sistema:

$$Qcs = ys$$

Que apresenta complexidade $O(n^3)$ para encontrar cs. Mas veja que, se Q for ortogonal:

$$Qcs = ys \implies Q^T Qcs = Q^T ys \implies cs = Q^T ys$$

Que apresenta complexidade $O(n^2)$ por conta do produto matriz-vetor.

Sejam Qs ortogonais: $Q_{0_{2^0 \times 2^0}}, Q_{1_{2^1 \times 2^1}}, \ldots, Q_{n_{2^n \times 2^n}}$

Seja f uma função tal que:

$$f(x) = \begin{cases} x & , \text{se } length(x) = 1 \\ \begin{bmatrix} \frac{f(x_1) + f(x_0)}{\sqrt{2}} \\ \frac{f(x_1) - f(x_0)}{\sqrt{2}} \end{bmatrix} & , \text{caso contrário.} \end{cases}$$

Tal que $x = \begin{bmatrix} x_1 \\ x_0 \end{bmatrix}$.

Todo: Por que f e Q fazem o mesmo?

f é um operador que faz o equivalente a Q. Ou seja, $f(cs) = Qcs = ys$.

Ao fazer $f(e_n)$, sendo e_n o vetor canônico da n^{a} coluna de matriz identidade, é possível visualizar que a matriz Q seria uma coluna composta somente de diversas instâncias de 1 e -1:

$$f\left(\begin{bmatrix}1\\0\\0\\0\end{bmatrix}\right) = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}, \quad f\left(\begin{bmatrix}0\\1\\0\\0\end{bmatrix}\right) = \begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}, \quad f\left(\begin{bmatrix}0\\0\\1\\0\end{bmatrix}\right) = \begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}, \quad f\left(\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right) = \begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}$$

Ou seja, para o caso (4×4) temos que:

$$Q = \begin{bmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{bmatrix}$$

Teorema 3.2.1. *A função f respeita as propriedades de uma matriz ortogonal. Ou seja, $\|x\|^2 = \|f(x)\|^2$ e $\langle x, y\rangle = \langle f(x), f(y)\rangle$ para quaisquer vetores x e y.*

Demonstração. **Quero mostrar que:** $\|x\|^2 = \|f(x)\|^2$.

Caso Base: por construção, quando $length(x) = 1$, temos $f(x) = x$, ou seja:

$$\|x\|^2 = x^2 = \|f(x)\|^2$$

Logo, a norma de x é preservada e o caso base está provado.

Hipótese de Indução: $\|F(c_1)\|^2 = \|c_1\|^2$ e $\|F(c_2)\|^2 = \|c_2\|^2$.

Ou seja, assumimos que f preserva a norma para vetores de comprimento menor.

Passo Indutivo: queremos mostrar que para um vetor $c = \begin{bmatrix}c_1\\c_2\end{bmatrix}$, temos:

$$\left\|c^2\right\| = \|f(c)\|^2$$

Pela definição de f, temos:

$$f(c) = \begin{bmatrix}\dfrac{f(c_1) + f(c_2)}{\sqrt{2}}\\[2ex] \dfrac{f(c_1) - f(c_2)}{\sqrt{2}}\end{bmatrix}.$$

Agora, vamos calcular $\|f(c)\|^2$:

$$\|f(c)\|^2 = \left\| \begin{bmatrix} \frac{f(c_1)+f(c_2)}{\sqrt{2}} \\ \frac{f(c_1)-f(c_2)}{\sqrt{2}} \end{bmatrix} \right\|^2 .$$

Expandindo, obtemos:

$$\|f(c)\|^2 = \frac{\|f(c_1)+f(c_2)\|^2}{2} + \frac{\|f(c_1)-f(c_2)\|^2}{2}.$$

Agora, usamos a identidade:

$$\|a+b\|^2 + \|a-b\|^2 = 2\|a\|^2 + 2\|b\|^2$$

Aplicando-a ao caso de $f(c_1)$ e $f(c_2)$:

$$\|f(c)\|^2 = \frac{2\|f(c_1)\|^2 + 2\|f(c_2)\|^2}{2} = \|f(c_1)\|^2 + \|f(c_2)\|^2$$

Pela hipótese de indução, sabemos que $\|f(c_1)\|^2 = \|c_1\|^2$ e $\|f(c_2)\|^2 = \|c_2\|^2$. Logo, temos:

$$\|f(c)\|^2 = \|c_1\|^2 + \|c_2\|^2 .$$

Como $\|c\|^2 = \|c_1\|^2 + \|c_2\|^2$, concluímos que:

$$\|f(c)\|^2 = \|c\|^2$$

Quero mostrar que: $\langle x, y\rangle = \langle f(x), f(y)\rangle$

Caso Base: Se $length(x) = length(y) = 1$, temos $f(x) = x$ e $f(y) = y$. Neste caso, o produto interno é claramente preservado e:

$$\langle f(x), f(y)\rangle = \langle f(x), f(y)\rangle = x \cdot y.$$

Hipótese de Indução: Suponha que, para vetores x_1, x_2, y_1 e y_2 de comprimento menor, temos:

$$\langle x_1, y_1\rangle = \langle f(x_1), f(y_1)\rangle \quad \text{e} \quad \langle x_2, y_2\rangle = \langle f(x_2), f(y_2)\rangle$$

Passo Indutivo: Agora, considere $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ e $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$. Pela definição de f, temos:

$$f(x) = \begin{bmatrix} \frac{f(x_1) + f(x_2)}{\sqrt{2}} \\ \frac{f(x_1) - f(x_2)}{\sqrt{2}} \end{bmatrix}, \quad f(y) = \begin{bmatrix} \frac{f(y_1) + f(y_2)}{\sqrt{2}} \\ \frac{f(y_1) - f(y_2)}{\sqrt{2}} \end{bmatrix}.$$

Agora, vamos calcular o produto interno $\langle f(x), f(y) \rangle$:

$$\langle f(x), f(y) \rangle = \left\langle \begin{bmatrix} \frac{f(x_1) + f(x_2)}{\sqrt{2}} \\ \frac{f(x_1) - f(x_2)}{\sqrt{2}} \end{bmatrix}, \begin{bmatrix} \frac{f(y_1) + f(y_2)}{\sqrt{2}} \\ \frac{f(y_1) - f(y_2)}{\sqrt{2}} \end{bmatrix} \right\rangle$$

Expandindo o produto interno, obtemos:

$$\begin{aligned} \langle f(x), f(y) \rangle &= \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} \left\langle \begin{bmatrix} f(x_1) + f(x_2) \\ f(x_1) - f(x_2) \end{bmatrix}, \begin{bmatrix} f(y_1) + f(y_2) \\ f(y_1) - f(y_2) \end{bmatrix} \right\rangle \\ &= \frac{1}{2}(f(x_1)f(y_1) + f(x_1)f(y_2) + f(x_2)f(y_1) + f(x_2)f(y_2) \\ &\quad + f(x_1)f(y_1) - f(x_1)f(y_2) - f(x_2)f(y_1) + f(x_2)f(y_2)) \\ &= \frac{1}{2}(2f(x_1)f(y_1) + 2f(x_2)f(y_2)) \\ &= f(x_1)f(y_1) + f(x_2)f(y_2) \\ &= \left\langle \begin{bmatrix} f(x_1) \\ f(x_2) \end{bmatrix}, \begin{bmatrix} f(y_1) \\ f(y_2) \end{bmatrix} \right\rangle \\ &= \langle f(x_1), f(y_1) \rangle + \langle f(x_2), f(y_2) \rangle \end{aligned}$$

Pela hipótese de indução, sabemos que:

$$\langle f(x_1), f(y_1) \rangle = \langle x_1, y_1 \rangle, \quad \langle f(x_2), f(y_2) \rangle = \langle x_2, y_2 \rangle$$

Logo, temos:

$$\begin{aligned} \langle f(x), f(y) \rangle &= \langle f(x_1), f(y_1) \rangle + \langle f(x_2), f(y_2) \rangle \\ &= \langle x_1, y_1 \rangle + \langle x_2, y_2 \rangle \\ &= \left\langle \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right\rangle \\ &= \langle x, y \rangle \end{aligned}$$

Portanto:

$$\langle f(x), f(y) \rangle = \langle x, y \rangle$$

Assim, mostramos que f preserva o produto interno e, portanto, é uma transformação ortogonal. □

Além disso, veja que se o tamanho do vetor de entrada é k, então a complexidade de f é $O(k \log_2(k))$:

Teorema 3.2.2. *A complexidade da função f, dado um vetor de tamanho k, é $O(k \log_2^k)$.*

Caso Base: para $k = 2^0$, temos que, pela definição de f:

$$f(x) = x, \text{ se } length(x) = 1$$

Logo, teremos uma operação $O(1)$, que respeita $O(k \log_2(k))$ pois:

$$k = 1 \implies 1 \log_2(1) = 0$$

Hipótese de Indução: para $k = 2^n$, a complexidade da função f, dado um vetor de tamanho k, é $O(k \log_2(k))$

Passo de Indução: para $k = 2^{n+1}$, temos:

$$\begin{aligned} O(k) &= O\left(\frac{2^{n+1}}{2}\right) + O\left(\frac{2^{n+1}}{2}\right) + \left(\frac{2^{n+1}}{2}\right) + \left(\frac{2^{n+1}}{2}\right) \\ &= 2 \cdot O\left(\frac{2^{n+1}}{2}\right) + 2\left(\frac{2^{n+1}}{2}\right) \end{aligned}$$

Tal que $O\left(\frac{2^{n+1}}{2}\right)$ seja por conta das chamadas recursivas $f(x_1) + f(x_0)$ e $f(x_1) - f(x_0)$.

E $2\left(\frac{2^{n+1}}{2}\right)$ seja por conta das operações de soma e subtração vetoriais. Veja então que:

$$\begin{aligned} O(k) &= 2 \cdot O\left(\frac{2^{n+1}}{2}\right) + 2\left(\frac{2^{n+1}}{2}\right) \\ &= 2O(2^n) + 2 \cdot 2^n \\ &= 2 \cdot O(2^n) + 2^{n+1} \end{aligned}$$

Pela Hipótese de Indução:

$$
\begin{aligned}
O(k) &= 2 \cdot 2^n log_2(2^n) + 2^{n+1} \\
&= 2^{n+1} log_2(2^n) + 2^{n+1} \\
&= 2^{n+1}(log_2(2^{n+1}) - log_2(2) + 1) \\
&= 2^{n+1}(log_2(2^{n+1}) \cancel{- log_2(2) + 1}) \\
&= 2^{n+1} log_2(2^{n+1})
\end{aligned}
$$

Visto que $k = 2^{n+1}$, então temos que f apresenta complexidade $2^k log_2(k)$.

Sendo assim, está provado que a complexidade da função f, dado um vetor de tamanho k, é $O(k \log_2(k))$.

□

Capítulo 4

Métodos Iterativos para Resolver Sistemas

4.1 Motivação para Métodos Iterativos para Resolver Sistemas

Todo: Mostrar o porquê resolver sistemas como a equação do calor é caro utilizando diferenças finitas (matriz esparsa)

Resolver sistemas lineares na forma $Ax = b$ requer que sejam utilizados métodos que apresentam complexidade $O(n^3)$, como a Eliminação Gaussiana, para sistemas exatos, ou alguma técnica de mínimos quadrados, para sistemas não exatos. Apesar de $O(n^3)$ não ser um problema em matrizes pequenas, conforme o valor de n varia, o número de computações necessárias para resolver o sistema cresce rapidamente. Além disso, outro problema é o alto consumo de memória do computador. Por exemplo:

Suponha que tenhamos uma barra de metal de tamanho 1 e sabemos a sua temperatura em ambos os extremos e queremos descobrir a temperatura no restante da barra. Podemos modelar o problema tal que a barra seja discretizada em n pontos.

Todo: legenda!

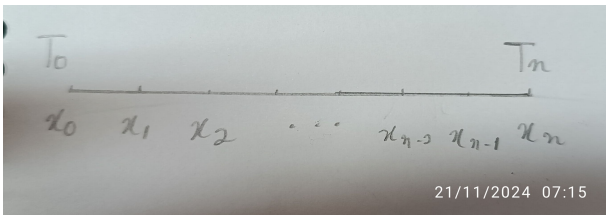

Figura 4.1: LEGENDA.

Sendo assim, podemos considerar que T_i é a temperatura no ponto x_i tal que x_i seja o ponto i de nossa barra e a temperatura de cada ponto seja considerada a média entre as temperaturas dos seus pontos adjacentes. Dessa forma, temos:

$$
\begin{aligned}
T_1 &= \frac{T_0 + T_2}{2} \\
T_i &= \frac{T_{i-1} + T_{i+1}}{2} \\
T_{n-1} &= \frac{T_{n-2} + T_n}{2}
\end{aligned}
$$

$$\forall i \in \{2, \ldots, n-2\}$$

Que é o mesmo que:

$$
\begin{aligned}
T_i &= \frac{T_{i-1} + T_{i+1}}{2} \\
2T_i &= T_{i-1} + T_{i+1} \\
2T_i - T_{i-1} - T_{i+1} &= 0
\end{aligned}
$$

$$\forall i \in \{1, \ldots, n-1\}$$

Ao variar o valor de i temos:

$$
\begin{aligned}
2T_1 - T_2 - T_0 &= 0 \\
2T_2 - T_3 - T_1 &= 0 \\
&\vdots \\
2T_i - T_{i-1} - T_{i+1} &= 0 \\
&\vdots \\
2T_{n-2} - T_{n-3} - T_{n-1} &= 0 \\
2T_{n-1} - T_{n-2} - T_n &= 0
\end{aligned}
$$

Mas sabemos os valores de T_0 e T_n (temperatura nos extremos da barra), logo, temos:

$$
\begin{aligned}
2T_1 - T_2 &= T_0 \\
2T_2 - T_3 - T_1 &= 0 \\
&\vdots \\
2T_i - T_{i-1} - T_{i+1} &= 0 \\
&\vdots \\
2T_{n-2} - T_{n-3} - T_{n-1} &= 0 \\
2T_{n-1} - T_{n-2} &= T_n
\end{aligned}
$$

Podemos colocar esse sistema no formato matriz-vetor:

$$\begin{bmatrix} 2 & -1 & & & 0 \\ -1 & \ddots & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & \ddots & -1 \\ 0 & & & -1 & 2 \end{bmatrix} \begin{bmatrix} T_1 \\ \vdots \\ T_i \\ \vdots \\ T_{n-1} \end{bmatrix} = \begin{bmatrix} T_0 \\ \vdots \\ 0 \\ \vdots \\ T_n \end{bmatrix}$$

Ao resolvermos esse sistema, teremos de fato uma solução para nosso problema estacionário da barra do calor. Ainda assim, veja que a matriz que fizemos é tridiagonal e extremamente esparsa.

A resolução do sistema linear exige o uso da Eliminação Gaussiana, um algoritmo com complexidade de $O(n^3)$, que é extremamente caro e não prático para matrizes tão grandes. Por isso, veremos outro método de complexidade menor para resolver sistemas lineares.

4.2 Método do Ponto Fixo

Podemos por meio de métodos numéricos semelhantes ao método do ponto fixo resolver o problema da alta complexidade de computação do sistema $Ax = b$. O método do ponto fixo diz que, dado um chute x_0 e uma função f contínua, é possível encontrar o ponto fixo x de f tal que $f(x) = x$ ao fazer $x^{k+1} = f(x^k)$ suficientes vezes. Com isso, podemos escrever

$$T_i = \frac{T_{i-1} + T_{i+1}}{2}$$

como

$$T_i^{k+1} = \frac{T_{i-1}^k + T_{i+1}^k}{2}$$

Veja que resolver essa iteração m vezes com $n-1$ equações faz com que o sistema apresente complexidade $O(m(n-1)) = O(mn)$.

Ainda assim, ao variar o valor de i, é possível escrever esse sistema de equações na forma $x^{k+1} = Mx^k + c$:

$$\begin{bmatrix} T_1^{k+1} \\ \vdots \\ T_i^{k+1} \\ \vdots \\ T_{n-1}^{k+1} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2} & & & 0 \\ \frac{1}{2} & \ddots & \frac{1}{2} & & \\ & \frac{1}{2} & 0 & \frac{1}{2} & \\ & & \frac{1}{2} & \ddots & \frac{1}{2} \\ 0 & & & \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} T_1^k \\ \vdots \\ T_i^k \\ \vdots \\ T_{n-1}^k \end{bmatrix} + \begin{bmatrix} \frac{T_0}{2} \\ \vdots \\ 0 \\ \vdots \\ \frac{T_n}{2} \end{bmatrix}$$

Perceba que esse sistema no formato $x^{k+1} = Mx^k + c$ realiza uma multiplicação matriz-vetor e uma soma entre vetores, operações que apresentam complexidade $O(n^2)$ e $O(n)$, respectivamente. Ou seja, se realizarmos m iterações, temos uma complexidade de $O(mn^2)$.

Definiremos o erro do nosso método como a diferença entre os valores encontramos em uma iteração do método e na iteração anterior. Ou seja:

$$e^{k+1} = x^{k+1} - x^k$$

Teorema 4.2.1. *Se $\forall i$ tal que λ_i é autovalor de M, $|\lambda_i| < 1$, então $\lim_{k\to\infty} e^{k+1} = 0$.*

Demonstração.

$$\begin{aligned}
\left\|e^{k+1}\right\| &= \left\|x^{k+1} - x^k\right\| \\
&= \left\|(Mx^k + c) - (Mx^{k-1} + c)\right\| \\
&= \left\|Mx^k + c - Mx^{k-1} - c\right\| \\
&= \left\|Mx^k - Mx^{k-1}\right\| \\
&= \left\|M(x^k - x^{k-1})\right\| \\
&= \left\|Me^k\right\|
\end{aligned}$$

Ou seja, temos que:

$$e^{k+1} = Me^k$$

E, com isso, temos que:

$$\begin{aligned}
e^{k+1} &= Me^k \\
&= MMe^{k-1} \\
&= MMMe^{k-2} \\
&\vdots \\
&= M^k e^0
\end{aligned}$$

Com isso, para que $\lim_{k\to\infty} e^{k+1} = 0$, precisamos de fato que M seja uma matriz que diminua o valor do vetor que a multiplica.

Suponha que M tenha pelo menos 2 autovalores λ_1 e λ_2 e seus 2 autovetores associados v e w linearmente independentes. Com isso, podemos escrever e^0 como:

$$e^0 = c_1 v + c_2 w$$
$$Me^0 = Mc_1 v + Mc_2 w$$

Mas como v e w são autovetores de M:

$$Me^0 = c_1 \lambda_1 v + c_2 \lambda_2 w$$
$$M^k e^0 = c_1 \lambda_1^k v + c_2 \lambda_2^k w$$

Mas como $|\lambda_1| < 1$ e $|\lambda_2| < 1$, então:

$$\begin{aligned}
\lim_{k\to\infty} e^{k+1} &= \lim_{k\to\infty} Me^k \\
&= \lim_{k\to\infty} M^k e^0 \\
&= \lim_{k\to\infty} c_1 \lambda_1^k v + c_2 \lambda_2^k w \\
&= \lim_{k\to\infty} c_1 \cancel{\lambda_1^k} v + c_2 \cancel{\lambda_2^k} w \\
&= 0
\end{aligned}$$

cotovelo-pca.jpg

Ou seja, está provado que $\lim_{k\to\infty} e^{k+1} = 0$. □

Corolário 4.2.1. *Sejam λ_i os autovalores de M e $\forall i$, $|\lambda_i| < 1$ e M tem autovetores linearmente independentes, então $x^{k+1} = Mx^k + c$ converge para o ponto fixo.*

Demonstração. Como provado anteriormente, $\lim_{k\to\infty} e^{k+1} = 0$, ou seja, $\lim_{k\to\infty} x^{k+1} - x^k = 0$, e $\lim_{k\to\infty} x^{k+1} = x^k$. Logo, o método converge para o ponto fixo. □

Note que, para valores de λ próximos de 1 em módulo, o método levará mais tempo para convergir para o ponto fixo, uma vez que λ^k decrescerá mais lentamente do que quando λ está próximo de 0.

4.3 Gauss-Jacobi e Gauss-Seidel

Mostramos que sabemos resolver sistemas na forma $x = Mx + c$ utilizando o método do ponto fixo. Dado um sistema $Ax = b$, o transformaremos em $x = Mx + c$.

Podemos escrever $Ax = b$ como:

$$\begin{bmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1n} \\ \vdots & \ddots & & & \vdots \\ a_{i1} & & a_{ij} & & a_{in} \\ \vdots & & & \ddots & \vdots \\ a_{n1} & \dots & a_{nj} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Também podemos escrever isso como:

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1j}x_i + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{i1}x_1 + \cdots + a_{ij}x_i + \cdots + a_{in}x_n &= b_i \\ &\vdots \\ a_{n1}x_1 + \cdots + a_{nj}x_i + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

Podemos isolar os valores a_{ij} da diagonal principal (ou seja, quando $i = j$) tal que:

$$\begin{aligned} x_1 &= \frac{\cdots - a_{ij}x_i - \cdots - a_{1n}x_n + b_1}{a_{11}} \\ &\vdots \\ x_i &= \frac{-a_{i1}x_1 - \cdots - a_{in}x_n + b_i}{a_{ij}} \\ &\vdots \\ x_n &= \frac{-a_{n1}x_1 - \cdots - a_{nj}x_i - \cdots + b_n}{a_{nn}} \end{aligned}$$

E, por fim, isolar a componente constante:

$$\begin{aligned} x_1 &= \frac{\cdots - a_{ij}x_i - \cdots - a_{1n}x_n}{a_{11}} + \frac{b_1}{a_{11}} \\ &\vdots \\ x_i &= \frac{-a_{i1}x_1 - \cdots - a_{in}x_n}{a_{ij}} + \frac{b_i}{a_{ij}} \\ &\vdots \\ x_n &= \frac{-a_{n1}x_1 - \cdots - a_{nj}x_i - \dots}{a_{nn}} + \frac{b_n}{a_{nn}} \end{aligned}$$

Veja que, dado um sistema nessa forma, podemos realizar um método semelhante ao método do ponto fixo tal que:

$$
\begin{aligned}
x_1^{k+1} &= \frac{\cdots - a_{ij}x_i^k - \cdots - a_{1n}x_n^k}{a_{11}} + \frac{b_1}{a_{11}} \\
&\vdots \\
x_i^{k+1} &= \frac{-a_{i1}x_1^k - \cdots - a_{in}x_n^k}{a_{ij}} + \frac{b_i}{a_{ij}} \\
&\vdots \\
x_n^{k+1} &= \frac{-a_{n1}x_1^k - \cdots - a_{nj}x_i^k}{a_{nn}} + \frac{b_n}{a_{nn}}
\end{aligned}
$$

Esse é o método de Gauss-Jacobi, que consiste em escrever o sistema $Ax = b$ na forma $x^{k+1} = Mx^k + c$ e realizar essas computações em ciclo até que o erro seja baixo o suficiente para uma condição de parada. Perceba que a primeira equação depende do valor do vetor x na iteração anterior, e o mesmo ocorre para a *iézima* e *nézima* equações. Sendo assim, esse método apresenta ótima complexidade computacional visto que as contas apresentam certa independência entre si no sentido de que podem ser executadas paralelamente.

Ainda assim, veja que após computar x_1^{k+1} (por exemplo), estamos utilizando x_1^k no cálculo de x_2^{k+1}. Visto isso, $\forall a < b$, podemos utilizar x_a^{k+1} no cálculo de x_b^{k+1}. Esse método, chamado de Gauss-Seidel, não permite paralelização como o anterior. Ainda assim, o erro converge com menos iterações para 0 visto que valores o mais atualizados possíveis estão sendo utilizados no cálculo de x.

$$
\begin{aligned}
x_1^{k+1} &= \frac{\cdots - a_{ij}x_i^k - \cdots - a_{1n}x_n^k}{a_{11}} + \frac{b_1}{a_{11}} \\
&\vdots \\
x_i^{k+1} &= \frac{-a_{i1}x_1^{k+1} - \cdots - a_{in}x_n^k}{a_{ij}} + \frac{b_i}{a_{ij}} \\
&\vdots \\
x_n^{k+1} &= \frac{-a_{n1}x_1^{k+1} - \cdots - a_{nj}x_i^{k+1}}{a_{nn}} + \frac{b_n}{a_{nn}}
\end{aligned}
$$

Capítulo 5

Sistemas Dinâmicos Lineares e Cadeias de Markov

5.1 Motivação para Sistemas Dinâmicos Lineares e Cadeias de Markov

Queremos modelar um problema tal que gostaríamos de prever a quantidade de pessoas em certas populações após determinado período de tempo.

Seja f a população de jovens e g a população de idosos.

Podemos modelar o sistema de duas maneiras:

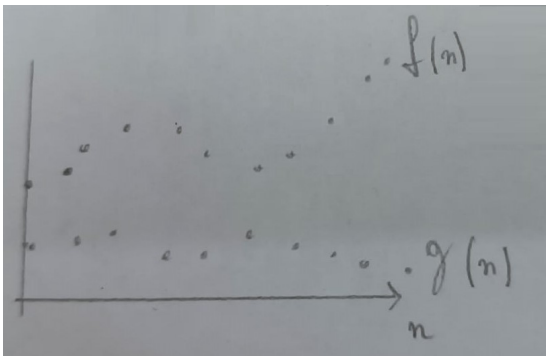

Figura 5.1: Modelagem do problema de forma discreta, sendo f e g sequências de pontos.

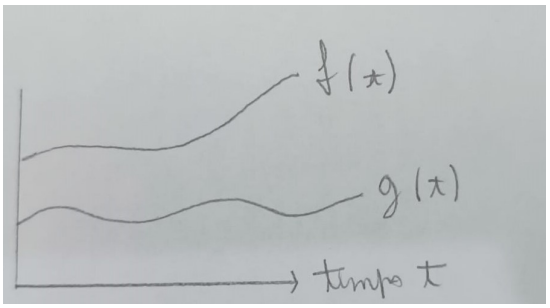

Figura 5.2: Modelagem do problema de forma contínua, sendo f e g funções contínuas em função do tempo

Podemos resolver o sistema contínuo a partir de conhecimentos do Cálculo ou Computação Numérica. Por exemplo, ao dizermos que $f'(t) = 0.7f(t)$, é possível que a solução do problema seja encontrada a partir de métodos analíticos como Fator Integrante, Equações Separáveis e Variação de Parâmetros ou métodos numéricos como Euler, Euler Melhorado e Runge-Kutta.

Podemos resolver o sistema discreto a partir de conhecimentos da Matemática Discreta. Por exemplo, ao dizermos que $f(n+1) = 3f(n)$ e $f(0) = 1$, é possível que a solução do problema seja encontrada a partir da resolução de recorrências.

5.2 Sistema Dinâmico Linear

Veja que, no nosso problema, é viável que f dependa de g e que g dependa de f, visto que a população de jovens depende necessariamente da população de idosos e vise-versa. Sendo assim, poderíamos ter um modelo que relacione as populações de jovens e idosos da seguinte maneira:

$$f(n+1) = a_{11}f(n) + a_{12}g(n)$$
$$g(n+1) = a_{21}f(n) + a_{22}g(n)$$

Conforme o valor de n varia, os valores de $f(n)$ e $g(n)$ também variarão, sendo assim, podemos visualizar esse sistema de equações como um sistema dinâmico linear.

Todo: legenda!

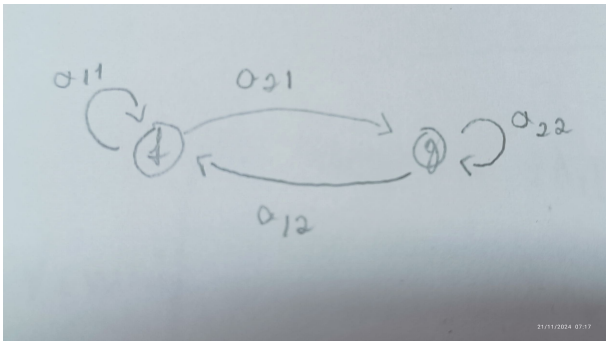

Figura 5.3: LEGENDA.

Uma forma de resolver esse sistema é a partir de um chute. Por exemplo:

$$f(n+1) = 3f(n) - g(n)$$
$$g(n+1) = -f(n) + 3g(n)$$

Seja $f(n) = \lambda^N$ e $g(n) = \lambda^N$ nossos chutes. Ao substituir na equação:

$$\lambda^{N+1} = 3\lambda^N - \lambda^N$$
$$\lambda^{N+1} = -\lambda^N + 3\lambda^N$$

Ao dividir ambos os lados das equações por λ^N (assumindo que $\lambda^N \neq 0$):

$$\lambda = 3 - 1 = 2$$
$$\lambda = -1 + 3 = 2$$

Ou seja, temos que $f(n) = g(n) = 2^N$.

Apesar desse método ser uma possível resolução para o problema, veja que ele não parece ser conveniente visto que tivemos que realizar um chute que desse certo e facilitasse nossas contas e também não temos a garantia de que a solução encontrada é única.

Sendo assim, dado o sistema

$$f(n+1) = a_{11}f(n) + a_{12}g(n)$$
$$g(n+1) = a_{21}f(n) + a_{22}g(n)$$

podemos expressá-lo a partir de um produto matriz-vetor:

$$\begin{bmatrix} f(n+1) \\ g(n+1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} f(n) \\ g(n) \end{bmatrix}$$

que, apesar de parecer conveniente, apresenta complexidade $O(mn^2)$, como mostrado em 4.2.

Veja que temos um vetor z como caso base do sistema dinâmico linear e:

$$z = \begin{bmatrix} f(0) \\ g(0) \end{bmatrix}$$

$$\begin{bmatrix} f(n) \\ g(n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^n \begin{bmatrix} f(0) \\ g(0) \end{bmatrix}$$

$$\begin{bmatrix} f(n) \\ g(n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^n z$$

Ou seja, aplicar a matriz n vezes em z é o mesmo que calcular $f(n)$ sendo z o caso base.

Assim como vimos em 4.2, podemos escrever o vetor z como combinação linear dos autovetores de A tal que:

$$
\begin{aligned}
z &= c_1 v + c_2 w \\
Az &= c_1 \lambda_1 v + c_2 \lambda_2 w \\
&\vdots \\
A^n z &= c_1 \lambda_1^n v + c_2 \lambda_2^n w
\end{aligned}
$$

Queremos encontrar c_1 e c_2 tais que:

$$
\begin{bmatrix} f(n) \\ g(n) \end{bmatrix} = c_1 \lambda_1^n v + c_2 \lambda_2^n w
$$

Dessa forma, podemos calcular os autovalores e autovetores de A:

v é dito autovetor de A se:

$$
Av = \lambda v
$$

Sendo λ um autovalor associado. Logo:

$$
\begin{aligned}
Av &= I\lambda v \\
Av - I\lambda v &= 0 \\
(A - I\lambda)v &= 0
\end{aligned}
$$

$A - I\lambda$ apresenta núcleo não trivial. Assim:

$$
\det(A - I\lambda) = 0
$$

Após resolver esse sistema, teremos os autovalores de A. Podemos usá-los para encontrar os autovetores v e w que satisfazem:

$$
(A - I\lambda_1)v = 0 \quad \text{e} \quad (A - I\lambda_2)w = 0
$$

Sendo assim, após termos os autovalores e autovetores, podemos encontrar c_1 e c_2 resolvendo o sistema:

$$
z = c_1 v + c_2 w
$$

Após termos os valores de c_1, c_2, v, w, λ_1 e λ_2, podemos calcular f e g sendo:

$$\begin{bmatrix} f(n) \\ g(n) \end{bmatrix} = c_1\lambda_1^n v + c_2\lambda_2^n w$$

Logo:

$$f(n) = c_1\lambda_1^n v_1 + c_2\lambda_2^n w_1$$
$$g(n) = c_1\lambda_1^n v_2 + c_2\lambda_2^n w_2$$

Um resultado interessante que é possível visualizarmos é a proporção entre as populações após uma determinada quantidade de tempo.

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{\text{Coeficiente da parte de maior ordem em } f(n)}{\text{Coeficiente da parte de maior ordem em } g(n)} = \frac{\alpha}{\beta}$$

E, com os valores de α e β podemos concluir:

- Para cada α jovens, há β idosos
- $\frac{\alpha}{\alpha+\beta}\%$ são jovens.
- $\frac{\beta}{\alpha+\beta}\%$ são idosos.

Essa proporção pode ser vista como a proporção dos autovetores associados aos maiores autovalores em módulo.

5.3 Exemplo de Solução

Dado o sistema:

$$f(n+1) = 3f(n) - g(n)$$
$$g(n+1) = -f(n) + 3g(n)$$

Gostaríamos de encontrar sua solução.

Podemos calcular seus autovalores utilizando $\det(A - \lambda I) = 0$:

$$\begin{aligned}
\det \begin{bmatrix} 3-\lambda & -1 \\ -1 & 3-\lambda \end{bmatrix} &= 0 \\
(3-\lambda)(3-\lambda) - 1 &= 0 \\
9 - 6\lambda + \lambda^2 - 1 &= 0 \\
\lambda^2 - 6\lambda + 8 &= 0
\end{aligned}$$

Que apresenta soluções em $\lambda_1 = 4$ e $\lambda_2 = 2$.

Para encontrar os autovetores, faremos $(A - \lambda I)v = 0$:

para $\lambda = 4$:

$$\begin{bmatrix} 3-4 & -1 \\ -1 & 3-4 \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}$$

Agora, resolvendo o sistema:

$$\begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Temos a equação:

$$-v_1 - v_2 = 0$$

Portanto, o autovetor associado a $\lambda_1 = 4$ é dado por:

$$v_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

para $\lambda = 2$:

$$\begin{bmatrix} 3-2 & -1 \\ -1 & 3-2 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Agora, resolvendo o sistema:

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Temos a equação:

$$w_1 - w_2 = 0$$

Portanto, o autovetor associado a $\lambda_2 = 2$ é dado por:

$$v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Agora, visto que $z = \begin{bmatrix} 1 \\ 3 \end{bmatrix} = c_1 v + c_2 w$:

$$\begin{bmatrix} 1 \\ 3 \end{bmatrix} = c_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Que resulta em $c_1 = 2$ e $c_2 = 1$. Sendo assim, podemos escrever:

$$\begin{aligned} f(n) = c_1\lambda_1^n v_1 + c_2\lambda_2^n w_1 &\implies f(n) = 2 \cdot 2^n 1 + 1 \cdot 4^n \cdot (-1) \\ g(n) = c_1\lambda_1^n v_2 + c_2\lambda_2^n w_2 &\implies g(n) = 2 \cdot 2^n 1 + 1 \cdot 4^n \cdot 1 \end{aligned}$$

Logo, $f(n) = 2 \cdot 2^n - 4^n$ e $g(n) = 2 \cdot 2^n + 4^n$.

Nesse caso, veja que:

$$\begin{aligned} \lim_{n\to\infty} \frac{f(n)}{g(n)} &= \lim_{n\to\infty} \frac{2 \cdot 2^n - 4^n}{2 \cdot 2^n + 4^n} \\ &= \lim_{n\to\infty} \frac{-4^n}{4^n} \\ &= \frac{-1}{1} \end{aligned}$$

Veja que $f(n) = 2^{n+1} - 4^n$ e $g(n) = 2^{n+1} + 4^n$ são de fato as equações que satisfazem o sistema:

$$\begin{aligned} f(n+1) &= 3f(n) - g(n) \\ g(n+1) &= -f(n) + 3g(n) \end{aligned}$$

Entretanto, observe que as expressões $f(n) = 2^{n+1} - 4^n$ e $g(n) = 2^{n+1} + 4^n$ não dizem muito sobre o nosso modelo. De forma semelhante, a equação:

$$f(x) = \alpha\pi^2 \text{sen}(\pi x) + \gamma\pi \cos(\pi x) + \beta \text{sen}(\pi x)$$

Também não revela muito, mas se reescrevemos como:

$$-\alpha u_{xx}(x) + \gamma u_x(x) + \beta u(x) = f(x)$$

Onde $u(x) = \text{sen}(\pi x)$, obtemos uma equação diferencial ordinária.

Analogamente, ao expressar as equações de f e g de forma que f não dependa de g, obtemos apenas uma equação que, embora correta, não oferece muita intuição sobre o modelo. No entanto, quando f e g são explicitamente correlacionados, estamos diante de um problema populacional.

5.4 Pagerank

Dado uma rede com n sites tais que todo site apresenta um *link* que direcione o usuário a outro site. Como é possível determinar o site mais importante?

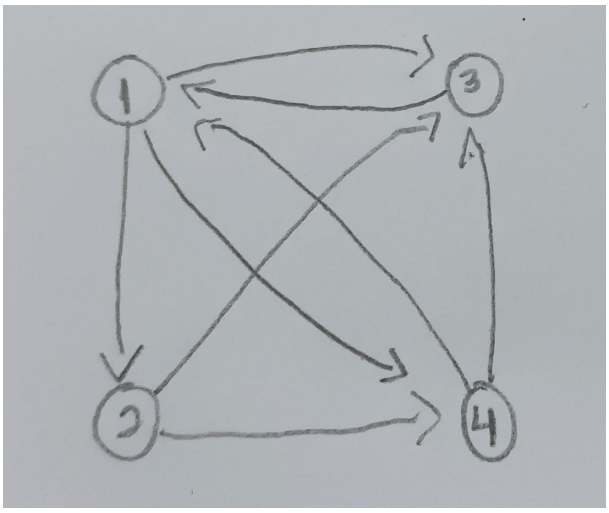

Figura 5.4: Representação do nosso problema em grafos.

Modelo 1: O site mais importante é o mais apontado.

Informalmente, podemos escolher um dos vértices e contar a quantidade de sites que apontam para ele. Ao fazer isso para todos os vértices, temos um algoritmo informal com complexidade $O(n^2)$ para estipular qual dos sites é o mais importante.

Ainda assim, essa modelagem apresenta uma falha: é possível que uma empresa crie diversos sites fantasmas[1] e faça-os redirecionar o usuário para um site desejado que não necessariamente é o mais importante de todos.

Modelo 2: O site mais importante é o mais apontado por sites importantes.

[1]Estamos considerando que um site é fantasma quando seu único intuito é burlar um algoritmo que julgue a importância de determinados sites (como o Pagerank).

Para evitar o problema acima, faremos com que um site seja considerado importante caso ele seja apontado por outros sites importantes. Ao definirmos nosso modelo dessa forma, estamos fazendo com que a definição de importância seja recursiva (mas sem apresentar caso base).

Podemos então — por meio dessa definição — escrever o sistema mostrado na figura 5.4 em equações. Seja x_i a importância do ponto i.

$$
\begin{aligned}
x_1 &= x_3 + x_4 \\
x_2 &= x_1 \\
x_3 &= x_1 + x_2 + x_4 \\
x_4 &= x_1 + x_2
\end{aligned}
$$

Logo:

$$
\begin{aligned}
0 &= -x_1 + x_3 + x_4 \\
0 &= x_1 - x_2 \\
0 &= x_1 + x_2 - x_3 + x_4 \\
0 &= x_1 + x_2 - x_4
\end{aligned}
$$

Ao escrever isso em um sistema matriz-vetor:

$$
\begin{bmatrix}
-1 & 0 & 1 & 1 \\
1 & -1 & 0 & 0 \\
1 & 1 & -1 & 1 \\
1 & 1 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ x_4
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
$$

Veja que:

$$
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ x_4
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
$$

É uma possível solução do sistema e que:

$$
det\left(
\begin{bmatrix}
-1 & 0 & 1 & 1 \\
1 & -1 & 0 & 0 \\
1 & 1 & -1 & 1 \\
1 & 1 & 0 & -1
\end{bmatrix}
\right) = -5
$$

Ou seja:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

é a única solução pro nosso sistema.

Veja que a modelagem acima está incorreta visto que todos os sites apresentam a mesma importância (0).

Além disso, nossa modelagem não parece ser justa visto que um site pode ser manipulado ou pago para apontar para outro. Além disso, sites poucos importantes podem parecer importantes.

Modelo 3: A importância de um site é dividida igualmente para os sites os quais aponta.

Todo: legenda!

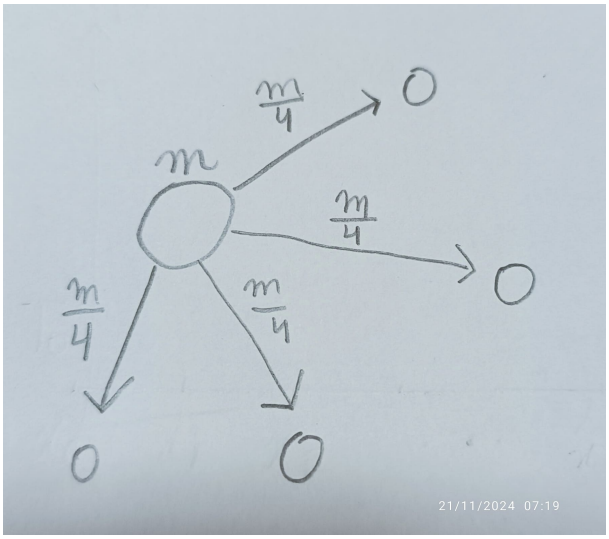

Figura 5.5: LEGENDA.

Sendo assim, caímos em um sistema de equações semelhante ao da modelagem anterior:

$$\begin{aligned} x_1 &= x_3 + \frac{x_4}{2} \\ x_2 &= \frac{x_1}{3} \\ x_3 &= \frac{x_1}{3} + \frac{x_2}{2} + \frac{x_4}{2} \\ x_4 &= \frac{x_1}{3} + \frac{x_2}{2} \end{aligned}$$

Logo:

$$
\begin{aligned}
0 &= -x_1 + x_3 + \frac{x_4}{2} \\
0 &= \frac{x_1}{3} - x_2 \\
0 &= \frac{x_1}{3} + \frac{x_2}{2} - x_3 + \frac{x_4}{2} \\
0 &= \frac{x_1}{3} + \frac{x_2}{2} - x_4
\end{aligned}
$$

Ao escrever isso em um sistema matriz-vetor:

$$
\begin{bmatrix}
-1 & 0 & 1 & \frac{1}{2} \\
\frac{1}{3} & -1 & 0 & 0 \\
\frac{1}{3} & \frac{1}{2} & -1 & \frac{1}{2} \\
\frac{1}{3} & \frac{1}{2} & 0 & -1
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

Mas dessa vez temos que:

$$
\det \begin{bmatrix}
-1 & 0 & \frac{1}{1} & \frac{1}{2} \\
\frac{1}{3} & -1 & 0 & 0 \\
\frac{1}{3} & \frac{1}{2} & -1 & \frac{1}{2} \\
\frac{1}{3} & \frac{1}{2} & 0 & -1
\end{bmatrix} = 0
$$

Ou seja, o sistema apresenta nenhuma ou infinitas soluções, mas como sabemos que o vetor com zeros é solução, então o sistema acima apresenta infinitas soluções.

Modelo 4: A importância de um site é dividida igualmente para os sites os quais aponta e a soma das importâncias precisa ser igual a 1.

Com essa condição a mais, teremos uma equação a mais no sistema tal que:

$$
\begin{bmatrix}
-1 & 0 & \frac{1}{1} & \frac{1}{2} \\
\frac{1}{3} & -1 & 0 & 0 \\
\frac{1}{3} & \frac{1}{2} & -1 & \frac{1}{2} \\
\frac{1}{3} & \frac{1}{2} & 0 & -1 \\
1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Dessa forma, o sistema apresenta solução única tal que:

$$
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
\approx
\begin{bmatrix} 0.387 \\ 0.129 \\ 0.29 \\ 0.193 \end{bmatrix}
$$

Apesar de termos encontrado uma solução que — para o exemplo de equação gerada pelo grafo da figura 5.4 — julga que os sites mais importantes sejam, respectivamente, o 1, 3, 4, 2;

temos o mesmo problema que encontramos na solução do sistema $Ax = b$ na seção 4.1: Temos uma matriz que, para o caso de uma rede de internet muito grande, apresenta grande esparsidade e complexidade $O(n^3)$ para resolver.

Modelo 5: A importância de um site é definida a partir do fluxo de pessoas que transitam por ele.

Faremos com que a probabilidade p de sair de um site i seja uniforme. Sendo assim, podemos montar o sistema da figura 5.4.

Podemos expressar essas equações na forma matriz-vetor tal que:

$$\begin{bmatrix} -1 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & -1 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & -1 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & -1 \end{bmatrix} \begin{bmatrix} p_1^k \\ p_2^k \\ p_3^k \\ p_4^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Mas veja que a resolução desse sistema apresenta complexidade $O(n^3)$. Além disso, perceba que de início as pessoas irão transitar pelos sites de maneira instável até que, em algum momento, o sistema se encontre em estado constante, de forma bem semelhante como o método de Gauss-Jacobi tem seu erro convergindo para 0. Sendo assim, podemos escrever esse sistema na forma $x^{k+1} = Mx^k + c$:

$$\begin{aligned} p_1^{k+1} &= p_3^k + \frac{1}{2}p_4^k \\ p_2^{k+1} &= \frac{1}{3}p_1^k \\ p_3^{k+1} &= \frac{1}{3}p_1^k + \frac{1}{2}p_1^k + \frac{1}{2}p_4^k \\ p_4^{k+1} &= \frac{1}{3}p_1^k + \frac{1}{2}p_1^k \end{aligned}$$

Que apresenta complexidade $O(knt)$, onde k = número de vizinhos, n = número de sites, e t = número de iterações. Perceba que t é um número que nós decidimos e k é necessariamente um número $\leq n - 1$, visto que um site não pode apontar para mais do que todos os sites do sistema exceto ele próprio.

Teorema 5.4.1. $\lim_{k\to\infty} p^{k+1} = p^k$

Demonstração. Na seção 4.2 definimos e^k como $x^{k+1} - x^k$ e provamos que:

$$\lim_{k\to\infty} e^k = 0$$

ou seja,

$$\lim_{k\to\infty} p^{k+1} - p^k = 0$$

que é o mesmo que:

$$\lim_{k\to\infty} p^{k+1} = p^k$$

Assim, provamos que $\lim_{k\to\infty} p^{k+1} = p^k$. □

Além disso, veja que: $p^{k+1} = Mp^k$, mas como $p^{k+1} = p^k$, então p^k é autovetor de M. Sabemos que $Av = \lambda v$, mas se $Mp^k = p^k$ então $\lambda = 1$ e p^k é seu autovetor associado.

Vimos então que aplicar uma matriz várias vezes em um vetor inicial faz com que os vetores resultantes estejam cada vez mais próximos (quando normalizados) à um vetor resultado. O Método da Potência é um algoritmo que nos garante que, dado uma matriz diagonalizável A, direção de seu autovetor pode ser encontrado ao aplicar a matriz diversas vezes em um vetor inicial:

$$\lim_{k\to\infty} \frac{A^k c}{\|A^k c\|} = \frac{v}{\|v\|}$$

Sendo A a matriz a qual desejamos encontrar seu autovetor v associado ao maior autovalor em módulo e c o vetor do chute inicial.

Visto que, pelo método da potência, o vetor p^k é levado ao autovetor associado ao autovalor de maior módulo e que p^k é o autovetor associado ao autovalor 1, então sabemos que o maior autovalor que a matriz M possui é necessariamente 1.

Caso o maior autovalor não fosse 1, uma alternativa seria normalizar o vetor que tende ao autovetor visto que geralmente estamos mais interessados na proporção entre as coordenadas do vetor resultante (autovetor) do que no seu valor exato. Veja que se $\lambda < 1$, então o vetor Mp^k estaria cada vez mais próximo ao vetor nulo do que p^k (Assim como na demonstração que realizamos para o cálculo do erro do método do ponto fixo na seção 4.2) e se $\lambda > 1$, então o vetor Mp^k cresceria em comparação à p^k, podendo assim ocasionar em *overflow*.

Todo: Terminar parágrafo acima

Todo: Devo incluir o exemplo de Fibonacci dado em sala? Talvez um exercício com isso seja melhor do que uma demonstração? Não sei!

Capítulo 6

Análise de Componente Principal e Redução de Dimensionalidade

6.1 Motivação para Redução de Dimensionalidade

Suponha que tenhamos uma imagem com resolução $m \times n$. Cada *pixel* dessa imagem é representado por uma estrutura contendo valores entre 0 e 255 para cada um dos três componentes: *Red*, *Green* e *Blue* (*RGB*). Assumindo que cada *pixel* tenha apenas esses três atributos (o que não é inteiramente verdade), podemos calcular que cada *pixel* ocupará $3 \cdot 8 = 24$ bits. Portanto, para uma imagem de resolução $m \times n$, o total de bits necessários será $24mn$, o que pode ser extremamente custoso quando m e n são grandes.

É claro que o custo de armazenamento da imagem está relacionado tanto à constante 24 quanto aos valores de m e n. No entanto, observe que nem todas as imagens precisam ser armazenadas com as 2^{8^3} combinações possíveis de cores que os 24 bits oferecem, nem na resolução nativa de $m \times n$.

Dessa forma, torna-se evidente que a redução da quantidade de cores e da resolução da imagem pode ser vantajosa em cenários onde essas variáveis são desnecessariamente grandes. Uma estratégia viável para a redução da quantidade de cores é agrupar os atributos que apresentam similaridade entre si (nesse caso, cores semelhantes), processo conhecido como Clusterização. A respeito da redução dimensional, veremos este conceito mais adiante.

Vale destacar que a redução de cores e de resolução mencionada aqui é apenas um exemplo de aplicação de técnicas como a Análise de Componentes Principais (*Principal Component Analysis* (PCA)). Essas técnicas não se limitam ao processamento de imagens; elas têm uma ampla gama de usos em diferentes áreas. Por exemplo, podem ser empregadas na compressão de dados, na identificação de padrões em conjuntos de dados multidimensionais, na redução de dimensionalidade para problemas de aprendizado de máquina e na melhoria do desempenho de algoritmos de reconhecimento de padrões. Abordaremos algumas dessas aplicações com o decorrer do material.

É evidente que quanto menor o nível de complexidade do sistema, maior será o erro, visto que teremos menos "bases" para representar todos os dados.

Todo: Legenda!

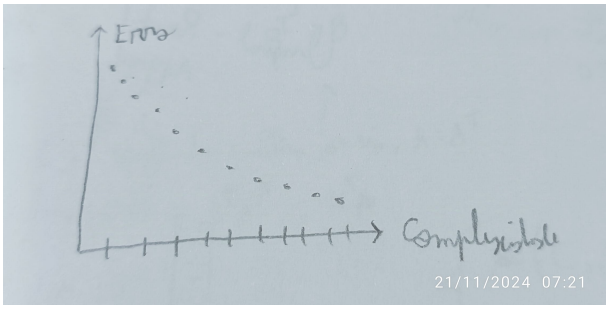

Figura 6.1: LEGENDA.

Outro exemplo é ao tentar descobrir a preferência de alguns clientes por determinados livros. Podemos muito bem classificar livros de acordo com suas categorias (como ação, comédia e romance) e também classificar as pessoas de acordo com suas preferências por cada gênero, colocar esses dados em matrizes e realizar uma multiplicação para descobrir a melhor relação entre livros e pessoas. Algo como:

$$\underset{\text{Pessoas}\times\text{Livros}}{\begin{bmatrix} \\ A \\ \\ \end{bmatrix}} = \underset{\text{Pessoas}\times\text{Gêneros}}{\begin{bmatrix} \\ B \\ \\ \end{bmatrix}} \underset{\text{Gêneros}\times\text{Livros}}{\begin{bmatrix} \\ C \\ \\ \end{bmatrix}}$$

Por mais que esse modelo funcione, perceba que ele apresenta um custo social muito grande no sentido que é complicado classificar livros e usuários por meio de seus gostos em diferentes gêneros. Também é questionável quais elementos colocar em Gênero dos Livros. Os livros e as pessoas são claros, mas o restante não. Além disso, veja que armazenar as matrizes B e C é caro realizar a multiplicação entre matrizes para gerar a matriz A toda vez que um dado for necessário visto que a multiplicação apresenta complexidade $O(n^3)$.

Uma possibilidade é expressar a matriz A por meio das matrizes B e C mas com uma escolha do tamanho da coluna que seja barato e conveniente para expressar os requisitos de nosso modelo bem o suficiente

Sendo assim, queremos expressar A tal que:

$$A_{u\times l} = B_{u\times n} \cdot C_{n\times l}$$

Mas, ainda assim, veja que ao decompor A em duas matrizes dado um valor arbitrário de n, não temos controle sobre o que o valor de n exatamente representa. Por exemplo: ao decompormos a matriz Pessoas $\times$ Livros e decidirmos selecionar 3 como o valor de n, o que exatamente os valores de B e C significarão? Veremos que é possível inferir seus significados dependendo do contexto e analisando os dados visualmente (Mas sem garantia da corretude da nossa inferência).

Perceba também que não faz sentido escolher valores grandes para n visto que estaremos representando uma matriz como produto de duas matrizes maiores (Que fere nossa motivação de otimizar os dados). Além disso, veja que quando $n = u$ teremos com que $A = B$ e $C = I$ ou $A = C$ e $B = I$ e o erro será 0.

6.2 Redução de Dimensão com PCA

Dado $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ sendo n seu tamanho, gostaríamos de visualizar esses pontos em uma reta para podermos, pela clusterização, agrupar diferentes dados em pequenos grupos para podermos julgarmos como semelhantes por algum critério.

Todo: legenda!

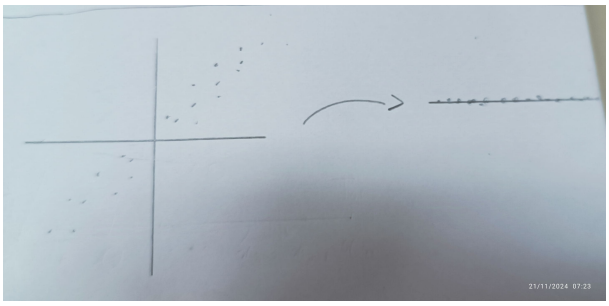

Figura 6.2: LEGENDA.

Modelo 1: Projetar os pontos no eixo x ou y.

Uma possibilidade é simplesmente projetar os pontos em um dos eixos. Essa estratégia funcionará para determinados padrões de pontos, mas veja que em funções que crescem muito rapidamente ou muito devagar o resultado não será benéfico visto que diversos dados diferentes serão considerados semelhantes.

Todo: Legenda!

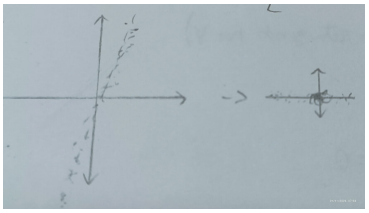

Figura 6.3: .

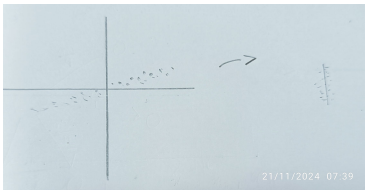

Figura 6.4: .

Modelo 2: Projetar os pontos na melhor reta que aproxima os pontos

Queremos que a melhor reta que aproxime os pontos seja aquela que apresenta a menor distância entre pontos e reta, diferente da aproximação que apresentamos na Regressão Polinomial (seção 1.2) em que a melhor reta seria aquela que minimizasse a distância vertical.

Todo: legenda!

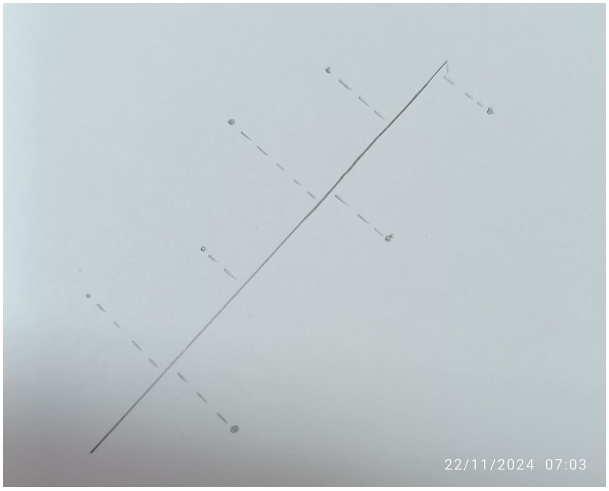

Figura 6.5: LEGENDA.

Todo: legenda!

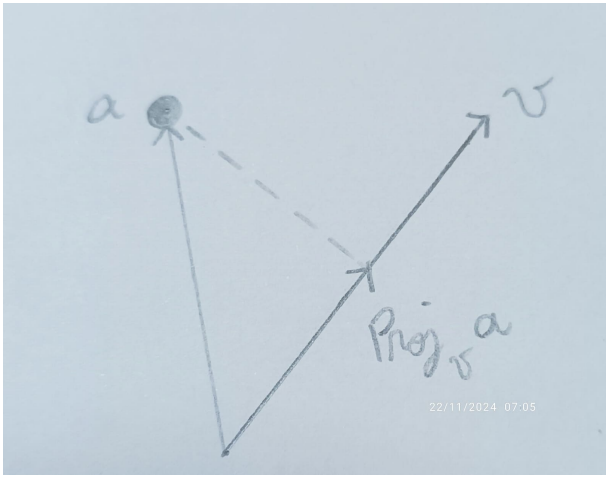

Figura 6.6: LEGENDA.

Para isso, precisamos antes encontrar a fórmula da projeção ortogonal e mostrar que a melhor aproximação de um ponto em em uma reta é sua projeção ortogonal.

Teorema 6.2.1. *A projeção de um vetor a em um vetor v é igual a* $\dfrac{\langle a, v\rangle}{\|v\|^2} v$.

Demonstração. A projeção de a em v terá a direção de v logo:

$$\text{proj}_a(v) = \lambda v, \text{ sendo } \lambda \in \mathbb{R} \tag{1}$$

Sendo assim, podemos ver que:

Todo: Legenda

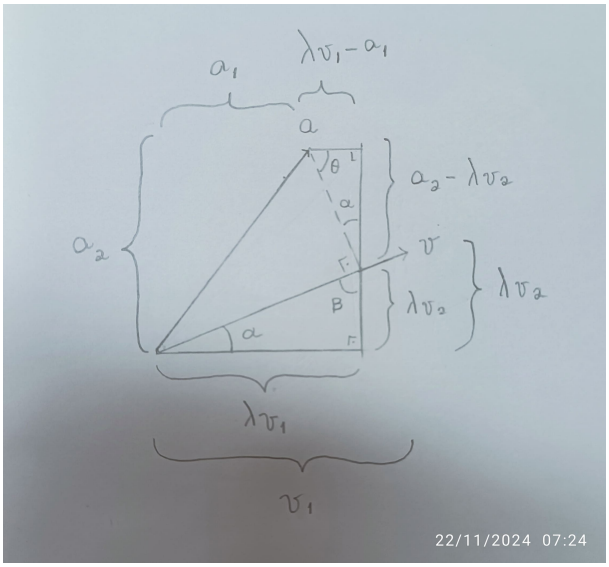

Figura 6.7: LEGENDA.

Veja que $\alpha + 90° + \beta = 180°$ e $\alpha + 90° + \theta = 180$. Ou seja, temos que:

$$\begin{aligned} \alpha + 90° + \beta &= \alpha + 90° + \theta \\ \theta &= \beta \end{aligned}$$

Logo temos que os triângulos inferior e lateral superior direito são semelhantes. Sendo assim, por semelhança de triângulos temos que:

$$\begin{aligned} \frac{\lambda v_2}{\lambda v_1} &= \frac{\lambda v_1 - a_1}{a_2 - \lambda v_2} \\ \frac{v_2}{v_1} &= \frac{\lambda v_1 - a_1}{a_2 - \lambda v_2} \\ v_2(a_2 - \lambda v_2) &= v_1(\lambda v_1 - a_1) \\ a_2 v_2 - \lambda v_2^2 &= \lambda v_1^2 - a_1 v_1 \\ \lambda v_1^2 + \lambda v_2^2 &= a_2 v_2 + a_1 v_1 \\ \lambda(v_1^2 + v_2^2) &= \langle a, v \rangle \\ \lambda(v_1^2 + v_2^2) &= \langle a, v \rangle \\ \lambda \sqrt{v_1^2 + v_2^2}^2 &= \langle a, v \rangle \\ \lambda \|v\|^2 &= \langle a, v \rangle \\ \lambda &= \frac{\langle a, v \rangle}{\|v\|^2} \end{aligned}$$

Substituindo em (1):

$$\text{proj}_a(v) = \frac{\langle a, v\rangle}{\|v\|^2} v$$

Portanto, provamos a projeção de um vetor a em um vetor v é igual a $\frac{\langle a, v\rangle}{\|v\|^2} v$. □

Teorema 6.2.2. *O vetor que apresenta a menor distância entre a e v é o vetor gerado pela diferença de a pela projeção de a em v.*

Demonstração. **Todo:** Legenda

Perceba que a menor distância de um ponto p qualquer para um vetor v resulta da diferença do vetor que representa o ponto p para algum vetor que apresenta a mesma direção de v.

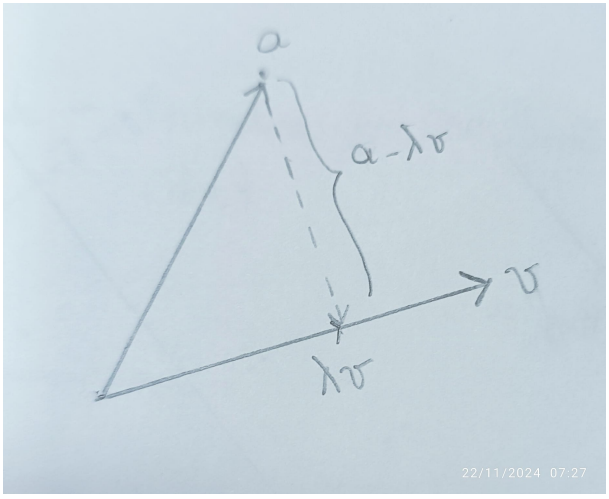

Figura 6.8: LEGENDA.

Queremos minimizar a função f tal que $f(\lambda) = \|a - \lambda v\|^2$ represente a distância entre a e λv. Sendo assim:

$$\text{Min}_\lambda\ f(\lambda) = \text{Min}_\lambda\ d(a, \lambda v) = \text{Min}_\lambda\ \|a - \lambda v\|^2$$

Mas sabemos que $\forall w \in \mathbb{R}^N$, $w^T w = \|w\|^2$, então:

$$\begin{aligned} \text{Min}_\lambda\ f(\lambda) &= \text{Min}_\lambda\ (a - \lambda v)^T (a - \lambda v) \\ &= \text{Min}_\lambda\ a^T a - 2\lambda a^T v + \lambda^2 v^T v \end{aligned}$$

Agora temos uma função de segundo grau em λ. Veja que a função apresenta concavidade para cima e, portanto, para encontrarmos o mínimo da função, podemos simplesmente encontrar o ponto em que sua derivada é igual a 0. Ou seja:

$$\begin{aligned}(a^Ta)' - (2\lambda a^Tv)' + (\lambda^2 v^Tv)' &= 0\\ 0 - 2a^Tv + 2\lambda v^Tv &= 0\\ 2\lambda v^Tv &= 2a^Tv\\ \lambda v^Tv &= a^Tv\\ \lambda &= \frac{a^Tv}{v^Tv}\\ \lambda &= \frac{\langle a, v\rangle}{\|v\|^2}\end{aligned}$$

Logo, O valor de λ que minimiza a distância entre a e λv é $\lambda = \frac{\langle a, v\rangle}{\|v\|^2}$. Veja também que isso significa que:

$$\lambda v = \frac{\langle a, v\rangle}{\|v\|^2} v$$

Que é a fórmula da projeção que encontramos em 6.2.1.

Logo, provamos que o vetor que representa a menor distância entre a e v é o vetor $a - \text{proj}_v(a)$. □

Teorema 6.2.3. *A menor distância entre um ponto a e a reta v é igual à* $\sqrt{\|a\|^2 - \left(\frac{\langle a, v\rangle}{\|v\|}\right)^2}$.

Demonstração. Provamos em 6.2.2 que o vetor na direção de v que minimiza a distância até a é:

$$\text{proj}_v(a) = \frac{\langle a, v\rangle}{\|v\|^2} v$$

Sendo assim, visto que $\text{proj}_v(a)$ representa uma projeção ortogonal (ou seja, apresenta ângulo reto), podemos utilizar o teorema de Pitágoras tal que:

Todo: legenda!

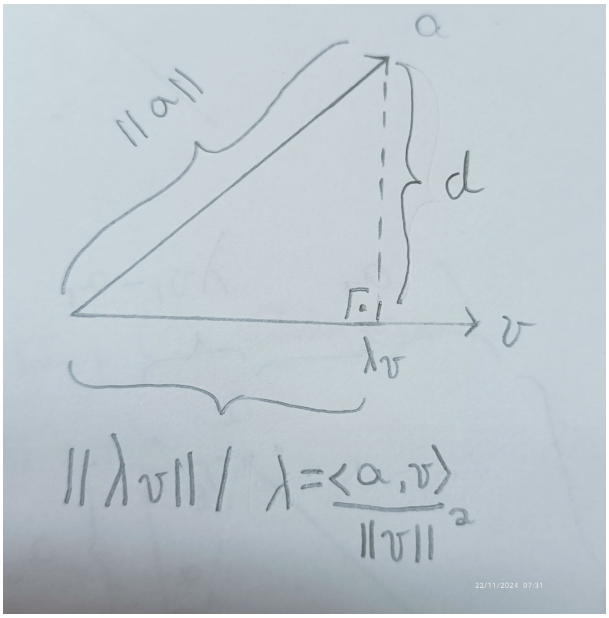

Figura 6.9: LEGENDA.

$$\|a\|^2 = d^2 + \left\| \frac{\langle a, v \rangle}{\|v\|^2} v \right\|^2$$

Sabemos que $\dfrac{\langle a, v \rangle}{\|v\|^2}$ é constante e também que $\forall k \in \mathbb{R}$, $\|kv\|^2 = k^2 \|v\|^2$, logo:

$$\begin{aligned}
\|a\|^2 &= d^2 + \left(\frac{\langle a, v \rangle}{\|v\|^2} \right)^2 \|v\|^2 \\
\|a\|^2 &= d^2 + \frac{\langle a, v \rangle^2}{\|v\|^4} \|v\|^2 \\
\|a\|^2 &= d^2 + \frac{\langle a, v \rangle^2}{\|v\|^2} \\
\|a\|^2 &= d^2 + \left(\frac{\langle a, v \rangle}{\|v\|} \right)^2 \\
d^2 &= \|a\|^2 - \left(\frac{\langle a, v \rangle}{\|v\|} \right)^2 \\
d &= \sqrt{\|a\|^2 - \left(\frac{\langle a, v \rangle}{\|v\|} \right)^2}
\end{aligned}$$

Assim, provamos que a menor distância entre um ponto a e uma reta v é $\sqrt{\|a\|^2 - \left(\dfrac{\langle a, v \rangle}{\|v\|} \right)^2}$. □

Todo: Provar que argmax f = argmin -f

Todo: Provar que argmin f + c = argmin f

Todo: Provar que $h(v)$ possui mínimo

Teorema 6.2.4. *$Argmin_x\ f(x) + c = Argmin_x\ f(x)$*

Teorema 6.2.5. *$Argmin_x\ -f(x) = Argmax_x\ f(x)$*

Teorema 6.2.6. *$h(v)$ possui mínimo global.*

Além disso, o erro de um ponto será calculado como sendo a distância dele para a reta que o aproxima seguindo a fórmula provada pelo teorema 6.2.3. Sendo assim, temos que o erro total do nosso método pode ser definido a partir de uma função $h(v)$ tal que:

$$h(v) = e_1^2 + \cdots + e_n^2$$

Teorema 6.2.7. *O erro do nosso método será o menor possível quando v for autovetor.*

Demonstração. Visto que, como provado em 6.2.3:

$$\forall i \in \{1, \ldots, n\},\ e_i = \sqrt{\|p_i\|^2 - \left(\frac{\langle p_i, v\rangle}{\|v\|}\right)^2}$$

Temos que:

$$e_i^2 = \|p_i\|^2 - \left(\frac{\langle p_i, v\rangle}{\|v\|}\right)^2$$

Logo:

$$h(v) = e_1^2 + \cdots + e_n^2 = \left(\|p_1\|^2 - \left(\frac{p_1^T v}{\|v\|}\right)^2\right) + \cdots + \left(\|p_n\|^2 - \left(\frac{p_n^T v}{\|v\|}\right)^2\right)$$

Ainda assim, veja que queremos minimizar o erro. Portanto, queremos resolver:

$$\text{Argmin}_v\ h(v) = \text{Argmin}_v\ \left(\|p_1\|^2 - \left(\frac{p_1^T v}{\|v\|}\right)^2\right) - \cdots - \left(\|p_n\|^2 - \left(\frac{p_n^T v}{\|v\|}\right)^2\right)$$

Veja que $\|p_i\|^2$, $\forall i \in \{1, \ldots, n\}$ é constante e não interfere em nossos cálculos. Logo, por 6.2.4, temos:

$$\text{Argmin}_v\ h(v) = \text{Argmin}_v\ \left(-\left(\frac{p_1^T v}{\|v\|}\right)^2 - \cdots - \left(\frac{p_n^T v}{\|v\|}\right)^2\right)$$

E que por 6.2.5, temos:

$$\begin{aligned}
\text{Argmin}_v\ h(v) &= \text{Argmax}_v\ \left(\left(\frac{p_1^T v}{\|v\|}\right)^2 + \cdots + \left(\frac{p_n^T v}{\|v\|}\right)^2\right) \\
&= \text{Argmax}_v\ \frac{1}{\|v\|^2}\left((p_1^T v)^2 + \cdots + (p_n^T v)^2\right) \\
&= \text{Argmax}_v\ \frac{1}{\|v\|^2}\left\|\begin{bmatrix} p_1^T v \\ \vdots \\ p_n^T v \end{bmatrix}\right\|^2 \\
&= \text{Argmax}_v\ \frac{1}{\|v\|^2}\left\|\begin{bmatrix} - & p_1^T & - \\ & \vdots & \\ - & p_n^T & - \end{bmatrix}\begin{bmatrix} | \\ V \\ | \end{bmatrix}\right\|^2 \\
&= \text{Argmax}_v\ \frac{1}{\|v\|^2}\|Av\|^2 \\
&= \text{Argmax}_v\ \frac{\|Av\|^2}{\|v\|^2}
\end{aligned}$$

Provamos em 6.2.6 que h possui mínimo global. Logo, por 6.2.5 sabemos que $-h$ possui máximo global. Sendo assim, para achar o valor de máximo de $-h$ podemos derivar a função e encontrar seus pontos críticos:

$$h_{vi}(v) = \left(\frac{\|Av\|^2}{\|v\|^2}\right)_{vi} = 0$$

Pela regra do quociente:

$$\frac{\|Av\|_{vi}^2 \|v\|^2 - \|Av\|^2 \|V\|_{vi}^2}{\|v\|^2} = 0$$

Como o resultado dessa conta é 0, o denominador não nos importa. Portanto, temos:

$$\begin{aligned}
\|Av\|_{vi}^2 \|v\|^2 - \|Av\|^2 \|V\|_{vi}^2 &= 0 \\
\|Av\|_{vi}^2 \|v\|^2 &= \|Av\|^2 \|V\|_{vi}^2
\end{aligned}$$

Calculando a parcela $\|Av\|_{vi}^2$:

$$
\begin{aligned}
\|Av\|_{vi}^2 &= ((Av)^T Av)_{vi} \\
&= (v^T A^T Av)_{vi} \\
&= \left(\begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \right)_{vi} \\
&= (v_1(c_{11}v_1 + c_{12}v_2) + v_2(c_{21}v_1 + c_{22}v_2))_{vi} \\
&= c_{11}v_1^2 + c_{12}v_1v_2 + c_{21}v_2v_1 + c_{22}v_2^2
\end{aligned}
$$

Derivando em v_1:

$$
(c_{11}v_1^2 + c_{12}v_1v_2 + c_{21}v_2v_1 + c_{22}v_2^2)_{v1} = 2c_{11}v_1 + 2c_{12}v_2
$$

Derivando em v_2:

$$
(c_{11}v_1^2 + c_{12}v_1v_2 + c_{21}v_2v_1 + c_{22}v_2^2)_{v1} = 2c_{22}v_2 + 2c_{21}v_1
$$

Calculando a parcela $\|V\|_{vi}^2$:

$$
\begin{aligned}
\|V\|_{vi}^2 &= (\sqrt{v_1^2 + v_2^2}^2)_{vi} \\
&= (v_1^2 + v_2^2)_{vi}
\end{aligned}
$$

Derivando em v_1:

$$
(v_1^2 + v_2^2)_{v1} = 2v_1
$$

Derivando em v_2:

$$
(v_1^2 + v_2^2)_{v2} = 2v_2
$$

Substituindo as derivadas em nossa equação principal:

$$
\begin{aligned}
2(c_{11}v_1 + c_{12}v_2) \|v\|^2 &= \|Av\|^2 2v_1 \\
2(c_{21}v_1 + c_{22}v_2) \|v\|^2 &= \|Av\|^2 2v_2
\end{aligned}
$$

$$
\begin{aligned}
(c_{11}v_1 + c_{12}v_2) &= \frac{\|Av\|^2}{\|v\|^2} v_1 \\
(c_{21}v_1 + c_{22}v_2) &= \frac{\|Av\|^2}{\|v\|^2} v_2
\end{aligned}
$$

Que é o mesmo que:

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{\|Av\|^2}{\|v\|^2} v$$

$$Cv = \frac{\|Av\|^2}{\|v\|^2} v$$

Ou seja, o $\text{Argmax}_v \ -h(v)$ é o v que resolve essa equação. Mas veja que:

$$Cv = \frac{\|Av\|^2}{\|v\|^2} v \implies v \text{ é autovetor e } \frac{\|Av\|^2}{\|v\|^2} \text{ autovalor.}$$

Ou seja, o $\text{Argmax}_v \ -h(v)$ é o v autovetor de C.

Logo, pelo teorema 6.2.5, sabemos que $\text{Argmax}_v \ -h(v) = \text{Argmin}_v \ h(v)$.

Logo, provamos que o erro do nosso método é minimizado quando v é autovetor de C. □

Todo: Provamos que v que minimiza o erro é autovetor, mas por que exatamente aquele associado ao maior autovalor em módulo?

Teorema 6.2.8. *Todo autovalor de C tal que $C = A^T A$ é do formato $\frac{\|Aw\|^2}{\|w\|^2}$ sendo w um autovetor.*

Demonstração.

$$\begin{aligned} \frac{\|Aw\|^2}{\|w\|^2} &= \frac{(Aw)^T Aw}{\|w\|^2} \\ &= \frac{w^T A^T Aw}{\|w\|^2} \\ &= \frac{w^T Cw}{\|w\|^2} \end{aligned}$$

Sabemos que $Cw = \lambda w$, logo:

$$
\begin{aligned}
&= \frac{w^T \lambda w}{\|w\|^2} \\
&= \lambda \frac{w^T w}{\|w\|^2} \\
&= \lambda \frac{\|w\|^2}{\|w\|^2} \\
&= \lambda
\end{aligned}
$$

Ou seja, todo autovalor de $C = A^T A$ é do formato $\frac{\|Aw\|^2}{\|w\|^2}$ sendo w um autovetor. □

Sendo assim, se C é de dimensão (2×2), dado autovalores λ_1 e λ_2 e autovetores w, $z \in \mathbb{R}^2$, temos que:

$$Cw = \lambda_1 w \quad Cz = \lambda_2 z$$

O que significa que:

$$\lambda_1 = \frac{\|Aw\|^2}{\|w\|^2}$$
$$\lambda_2 = \frac{\|Az\|^2}{\|z\|^2}$$

Teorema 6.2.9. *Os autovalores de uma matriz C tal que $C = A^T A$ são sempre positivos e reais.*

Demonstração. Provamos em 6.2.8 que os autovalores de uma matriz $C = A^T A$ respeitam a igualdade:

$$\forall i \in \{1, \ldots, n\}, \quad \lambda_i = \frac{\|Aw_i\|^2}{\|w_i\|^2}$$

Sendo λ_i qualquer autovalor de C e w_i seu autovetor associado. Sendo assim, todos os autovalores de C são necessariamente a razão entre duas normas ao quadrado. Visto que a norma de um vetor é necessariamente um número real e que a razão entre dois números reais é um número real, então temos que:

$$\frac{\|Aw_i\|^2}{\|w_i\|^2} \in \mathbb{R}$$

Além disso, visto que $\|Aw_i\|^2$ é um número positivo (por conta do expoente positivo) e que $\|w_i\|^2$ também é positivo, temos uma razão entre números positivos que, por sua vez, também é positiva.

Logo, provamos que os autovalores da matriz C tal que $C = A^T A$ são sempre positivos e reais. □

Teorema 6.2.10. *Seja A uma matriz $(n \times n)$. $\forall x, y \in \mathbb{R}^n$, $\langle Ax, y\rangle = \langle x, Ay\rangle \iff A^T = A$.*

Demonstração. Precisamos mostrar ambas as implicações.

Quero mostrar que: $A^T = A \implies \langle Ax, y\rangle = \langle x, Ay\rangle$:

$$\begin{aligned}\langle Ax, y\rangle &= (Ax)^T y \\ &= x^T A^T y \\ &= x^T Ay \\ &= \langle x, Ay\rangle\end{aligned}$$

Provamos que $A^T = A \implies \langle Ax, y\rangle = \langle x, Ay\rangle$.

Quero mostrar que: $\langle Ax, y\rangle = \langle x, Ay\rangle \implies A^T = A$

$$A_{ij} = \langle e_i, Ae_j\rangle$$

Temos que $\langle e_i, Ae_j\rangle = \langle Ae_i, e_j\rangle$, logo:

$$\langle Ae_i, e_j\rangle = A_{ji}$$

Provamos que $\langle Ax, y\rangle = \langle x, Ay\rangle \implies A^T = A$

Dessa forma, provamos que $\langle Ax, y\rangle = \langle x, Ay\rangle \iff A^T = A$. □

Teorema 6.2.11. *Se C é uma matriz simétrica, então seus autovetores associados a autovalores distintos são ortogonais entre si.*

Demonstração. Sejam $\lambda_1, \ldots, \lambda_n$ autovalores de C com $\lambda_i \neq \lambda_j$ para $i \neq j$ tal que $i, j \in \{1, \ldots, n\}$, e que $v_1, \ldots, v_n$ sejam os autovetores correspondentes a $\lambda_1, \ldots, \lambda_n$, respectivamente. Assim, temos:

$$Cv_i = \lambda_i v_i$$

Como C é simétrica, temos que $C = C^T$ e que, pelo teorema 6.2.10, sabemos que:

$$\langle Cv_i, v_j \rangle = \langle v_i, Cv_j \rangle$$

Que é o mesmo que:

$$(Cv_i)^T v_j = v_i^T (Cv_j)$$

Logo, temos:

$$\begin{aligned} v_i^T (Cv_j) &= v_i^T (\lambda_j v_j) = \lambda_j (v_i^T v_j), \\ (Cv_i)^T v_j &= (\lambda_i v_i)^T v_j = \lambda_i (v_i^T v_j). \end{aligned}$$

Como $C = C^T$, temos que $v_i^T (Cv_j) = (Cv_i)^T v_j$. Portanto:

$$\lambda_j (v_1^T v_j) = \lambda_1 (v_1^T v_j).$$

Como $\lambda_i \neq \lambda_j$, temos que $v_i^T v_j = 0$. Assim, v_i e v_j são ortogonais.

Portanto, mostramos que os autovetores associados a autovalores distintos de uma matriz simétrica são ortogonais. □

Todo: Por que $\frac{\|Av\|^2}{\|v\|^2}$ é igual ao somatório das sombras?

Visto que $\frac{\|Av\|^2}{\|v\|^2}$ representa a soma das projeções (ou "sombras") de um vetor sobre as direções principais, precisamos demonstrar que, $\forall w \in \mathbb{R}^3$, a função que determina a soma das sombras atinge seu valor máximo quando w está orientado na direção do maior autovalor. Além disso, ao removermos o componente correspondente a esse maior autovalor, o novo valor máximo dessa função será dado pelo segundo maior autovalor.

Teorema 6.2.12. *Se $v_1, \ldots, v_n$ forem vetores ortonormais, então: $\forall \alpha_i \in \mathbb{R}$, $i \in \{1, \ldots, n\}$, $\|\sum_{i=1}^n \alpha_i v_i\|^2 = \sum_{i=1}^n \alpha_i^2$.*

Demonstração. Sabemos que $\|w\|^2 = \langle w, w \rangle$, logo:

$$\left\| \sum_{i=1}^n \alpha_i v_i \right\|^2 = \left\langle \sum_{i=1}^n \alpha_i v_i, \sum_{j=1}^n \alpha_j v_j \right\rangle$$

Sabemos também que $\langle \lambda w, u \rangle = \lambda \langle w, u \rangle$, sendo w e u vetores quaisquer e λ uma constante. Sendo assim:

$$\left\langle \sum_{i=1}^{n} \alpha_i v_i, \sum_{j=1}^{n} \alpha_j v_j \right\rangle = \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{n} \alpha_j \langle v_i, v_j \rangle$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \langle v_i, v_j \rangle$$

Mas veja que, como v_i e v_j são ortogonais $\forall i \neq j$, então temos que:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \langle v_i, v_j \rangle = \sum_{i=1}^{n} \alpha_i \alpha_i \langle v_i, v_i \rangle$$
$$= \sum_{i=1}^{n} \alpha_i^2 \langle v_i, v_i \rangle$$

E temos que $\langle v_i, v_i \rangle = \|v_i\|^2$ como v_i é normal, temos que $\|v_i\|^2 = 1$, logo:

$$\sum_{i=1}^{n} \alpha_i^2 \langle v_i, v_i \rangle = \sum_{i=1}^{n} \alpha_i^2 \|v_i\|^2 = \sum_{i=1}^{n} \alpha_i^2$$

Logo, concluímos que se $v_1, \ldots, v_n$ forem vetores ortonormais, então: $\forall \alpha_i \in \mathbb{R}, i \in \{1, \ldots, n\}$, $\|\sum_{i=1}^{n} \alpha_i v_i\|^2 = \sum_{i=1}^{n} \alpha_i^2$. □

Teorema 6.2.13. *Podemos escrever* $\dfrac{\|Aw\|^2}{\|w\|^2}$ *como* $\dfrac{\alpha_1^2 \lambda_1 + \cdots + \alpha_n^2 \lambda_n}{\alpha_1^2 + \cdots + \alpha_n^2}$, *sendo* $\alpha_i \in \mathbb{R} \forall i \in \{1, \ldots, n\}$.

Demonstração. Por 6.2.7 sabemos que o erro é minimizado quando v é autovetor, mas faltou provar que este é o autovetor associado ao maior autovalor. Além disso, provamos em 6.2.9 que o autovalor é necessariamente positivo. Sendo assim, não precisamos nos preocupar com autovalores em módulo.

Provamos em 6.2.8 que todo autovalor de C tem formato $\dfrac{\|Aw\|^2}{\|w\|^2}$, sendo assim, temos que:

$$\frac{\|Aw\|^2}{\|w\|^2} = \frac{\langle Aw, Aw \rangle}{\|w\|^2}$$
$$= \frac{(Aw)^T Aw}{\|w\|^2}$$
$$= \frac{w^T A^T Aw}{\|w\|^2}$$
$$= \frac{w^T Cw}{\|w\|^2}$$

Pelo teorema 6.2.11, sabemos que os autovetores (associados a autovalores diferentes) de C são todos ortogonais entre si. Logo, eles são linearmente independentes e podemos escrever qualquer vetor do subespaço gerado pelos autovetores de C como combinação linear dos autovetores, logo, dado um vetor w qualquer:

$$w = \alpha_1 v_1 + \cdots + \alpha_n v_n$$

Tal que $v_1, \ldots, v_n$ são autovetores ortogonais de C normalizados (ortonormais).

Sendo assim, temos:

$$\frac{w^T C w}{\|w\|^2} = \frac{(\alpha_1 v_1 + \cdots + \alpha_n v_n)^T C (\alpha_1 v_1 + \cdots + \alpha_n v_n)}{\|\alpha_1 v_1 + \cdots + \alpha_n v_n\|^2}$$

Sabemos também que $\forall i \in \{1, \ldots, n\}$, $Cv_i = \lambda_i v_i$, então, temos:

$$\frac{(\alpha_1 v_1 + \cdots + \alpha_n v_n)^T (\alpha_1 \lambda_1 v_1 + \cdots + \alpha_n \lambda_n v_n)}{\|\alpha_1 v_1 + \cdots + \alpha_n v_n\|^2}$$

Que é o mesmo que:

$$\frac{\sum_{i=1}^{n} \alpha_i v_i^T \left(\sum_{j=1}^{n} \alpha_j \lambda_j v_j \right)}{\|\alpha_1 v_1 + \cdots + \alpha_n v_n\|^2}$$

Mas como $v_1, \ldots, v_n$ são ortogonais entre si, pelo teorema 6.2.11, então $v_i^T v_j = 0$ para $i \neq j$ e $v_i^T v_i = \|v_i\|^2$. Logo, substituindo na equação obtemos:

$$\frac{\alpha_1^2 \lambda_1 \|v_1\|^2 + \cdots + \alpha_n^2 \lambda_n \|v_n\|^2}{\|\alpha_1 v_1 + \cdots + \alpha_n v_n\|^2}$$

Pelo teorema 6.2.12, temos que $\|\alpha_1 v_1 + \cdots + \alpha_n v_n\|^2 = \alpha_1^2 + \cdots + \alpha_n^2$, logo:

$$\frac{\alpha_1^2 \lambda_1 \|v_1\|^2 + \cdots + \alpha_n^2 \lambda_n \|v_n\|^2}{\alpha_1^2 + \cdots + \alpha_3^2}$$

Visto que $v_1, \ldots, v_3$ são ortonormais, temos que $\forall i \in \{1, \ldots, n\}$, $\|v_i\| = 1$. Portanto, as normas ao quadrado também serão iguais a 1. Assim, temos:

$$\frac{\alpha_1^2 \lambda_1 + \cdots + \alpha_n^2 \lambda_n}{\alpha_1^2 + \cdots + \alpha_n^2}$$

Sendo assim, provamos que $\frac{\|Aw\|^2}{\|w\|^2} = \frac{\alpha_1^2\lambda_1 + \cdots + \alpha_n^2\lambda_n}{\alpha_1^2 + \cdots + \alpha_n^2}$, sendo $\alpha_i \in \mathbb{R}$ e $\forall i \in \{1, \ldots, n\}$.

□

Teorema 6.2.14. *O erro do nosso método é minimizado quando v for o autovetor associado ao maior autovalor.*

Demonstração. Visto que queremos mostrar que v associado ao maior autovalor maximiza as sombras (e minimiza o erro), queremos provar que:

$$\frac{\|Av_1\|^2}{\|v_1\|^2} \geq \frac{\|Aw_i\|^2}{\|w_i\|^2} \quad \forall i \in \{2, \ldots, n\}$$

Sendo v_1 o autovetor associado ao maior autovalor. Provamos em 6.2.7 que o erro é minimizado quando v é autovetor. Sendo assim, não precisamos nos preocupar com o lado direito da equação podendo ser qualquer outro vetor a não ser autovetor de A^TA.

Por 6.2.13 podemos reescrever essas equações tais que:

$$\begin{aligned} \frac{\|Av_1\|^2}{\|v_1\|^2} &\geq \frac{\|Aw_i\|^2}{\|w_i\|^2} \\ \frac{\alpha_1^2\lambda_1 + \cdots + \alpha_n^2\lambda_n}{\alpha_1^2 + \cdots + \alpha_n^2} &\geq \frac{\beta_1^2\lambda_1 + \cdots + \beta_n^2\lambda_n}{\beta_1^2 + \cdots + \beta_n^2} \end{aligned}$$

Provamos em 6.2.8 que $\frac{\|Av_1\|^2}{\|v_1\|^2} = \lambda_1$, ou seja, temos que $\alpha_2, \ldots, \alpha_n = 0$ e então:

$$\begin{aligned} \frac{\alpha_1^2\lambda_1}{\alpha_1^2} &\geq \frac{\beta_1^2\lambda_1 + \cdots + \beta_n^2\lambda_n}{\beta_1^2 + \cdots + \beta_n^2} \\ \lambda_1 &\geq \frac{\beta_1^2\lambda_1 + \cdots + \beta_n^2\lambda_n}{\beta_1^2 + \cdots + \beta_n^2} \\ (\beta_1^2 + \cdots + \beta_n^2)\lambda_1 &\geq \beta_1^2\lambda_1 + \cdots + \beta_n^2\lambda_n \\ \beta_1^2\lambda_1 + \cdots + \beta_n^2\lambda_1 &\geq \beta_1^2\lambda_1 + \cdots + \beta_n^2\lambda_n \end{aligned}$$

visto que $\forall i \in \{1, \ldots, n\}$, $\lambda_1 \geq \lambda_i$, então temos que $\frac{\|Av_1\|^2}{\|v_1\|^2} \geq \frac{\|Av_i\|^2}{\|v_i\|^2}$.

Sendo assim, temos que λ_1 é o máximo de $\frac{\|Av_i\|^2}{\|v_i\|^2}$ e provamos que o erro é minimizado quando v é o autovetor associado ao maior autovalor. □

6.3 Exemplo de Redução de Dimensionalidade

Dado pontos $p_1 = (0, 3)$, $p_2 = (4, 5)$ e $p_3 = (0, 0)$, como aproximá-los da melhor maneira possível no $\mathbb{R}^1$? Como encontrar a melhor reta?

Todo: legenda!

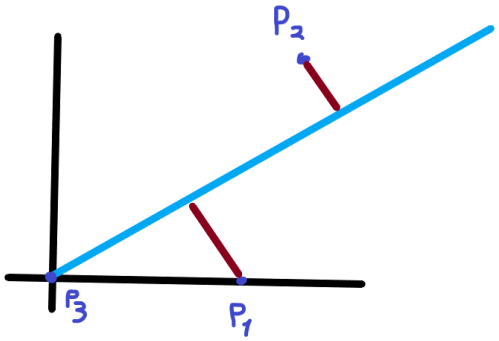

Figura 6.10: LEGENDA.

Podemos colocar os pontos em uma matriz tal que:

$$A = \begin{bmatrix} - & p_1 & - \\ - & p_2 & - \\ - & p_3 & - \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 4 & 5 \\ 0 & 0 \end{bmatrix}$$

Escolhemos que os vetores da matriz A estejam deitados pois assim sua dimensão é (3×2), e queremos que a matriz C seja a menor possível:

$$C = A^T A = \begin{bmatrix} 3 & 4 & 0 \\ 0 & 5 & 0 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 4 & 5 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix}$$

Sabemos que precisamos encontrar seus autovalores pois vimos em 6.2 que a projeção dos pontos na reta serão o mais espalhados (terão a maior sombra) quando o erro for mínimo, que ocorre quando utilizamos o autovetor associado ao maior autovalor como o primeiro componente principal. Sendo assim, precisamos calcular os autovalores. Mostramos em [ALGUM LUGAR][1] que é possível calcular os autovalores fazendo:

Todo: Verificar o footnote!

[1] Essa prova está escrita na seção 5.2, mas devemos fazer uma seção apenas para isso? Não faz sentido essa prova estar nessa seção

$$det(A - \lambda I) = 0$$

Logo:

$$\begin{aligned} \det \begin{bmatrix} 25-\lambda & 20 \\ 20 & 25-\lambda \end{bmatrix} &= 0 \\ (25-\lambda)(25-\lambda) - (20 \cdot 20) &= 0 \\ 625 - 25\lambda + \lambda^2 &= 0 \\ \lambda_1 = 45 \text{ e } \lambda_2 &= 5 \end{aligned}$$

Sendo assim, o autovetor associado ao maior autovalor é:

$$\begin{aligned} \begin{bmatrix} 25-45 & 20 \\ 20 & 25-45 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ -20w_1 + 20w_2 &= 0 \\ 20w_1 - 20w_2 &= 0 \end{aligned}$$

Que significa que $w_1 = w_2$. Logo, temos:

$$k \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{, sendo } k \in \mathbb{R}$$

Sendo assim, temos a direção da reta que projetaremos nossos pontos.

Projeção de p_1 na reta:

$$\begin{aligned} \frac{P_1^T v}{\|v\|^2} v &= \left(\frac{\begin{bmatrix} 3 \\ 0 \end{bmatrix}^T \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|^2} \right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \frac{3}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} \end{aligned}$$

Projeção de p_2 na reta:

$$\begin{aligned}\frac{P_2^T v}{\|v\|^2} v &= \left(\frac{\begin{bmatrix}4\\5\end{bmatrix}^T \begin{bmatrix}1\\1\end{bmatrix}}{\left\| \begin{bmatrix}1\\1\end{bmatrix} \right\|^2} \right) \begin{bmatrix}1\\1\end{bmatrix} \\ &= \frac{9}{2} \begin{bmatrix}1\\1\end{bmatrix} \\ &= \begin{bmatrix}4.5\\4.5\end{bmatrix}\end{aligned}$$

Projeção de p_3 na reta:

$$\begin{aligned}\frac{P_3^T v}{\|v\|^2} v &= \left(\frac{\begin{bmatrix}0\\0\end{bmatrix}^T \begin{bmatrix}1\\1\end{bmatrix}}{\left\| \begin{bmatrix}1\\1\end{bmatrix} \right\|^2} \right) \begin{bmatrix}1\\1\end{bmatrix} \\ &= \frac{0}{2} \begin{bmatrix}1\\1\end{bmatrix} \\ &= \begin{bmatrix}0\\0\end{bmatrix}\end{aligned}$$

Veja então que encontramos que as aproximações para nossos pontos originais que minimizam os erros são:

$$\begin{bmatrix}3\\0\end{bmatrix} \approx \begin{bmatrix}1.5\\1.5\end{bmatrix} \qquad \begin{bmatrix}4\\5\end{bmatrix} \approx \begin{bmatrix}4.5\\4.5\end{bmatrix} \qquad \begin{bmatrix}0\\0\end{bmatrix} \approx \begin{bmatrix}0\\0\end{bmatrix}$$

Que é o mesmo que:

$$\begin{bmatrix}3\\0\end{bmatrix} \approx 1.5 \begin{bmatrix}1\\1\end{bmatrix} \qquad \begin{bmatrix}4\\5\end{bmatrix} \approx 4.5 \begin{bmatrix}1\\1\end{bmatrix} \qquad \begin{bmatrix}0\\0\end{bmatrix} \approx 0 \begin{bmatrix}1\\1\end{bmatrix} \tag{1}$$

Esses valores que encontramos 1.5, 4.5 e 0 são chamados de endereços. Esses valores correspondem à posição no $\mathbb{R}^1$ que representa cada ponto de entrada do problema.

Todo: legenda!

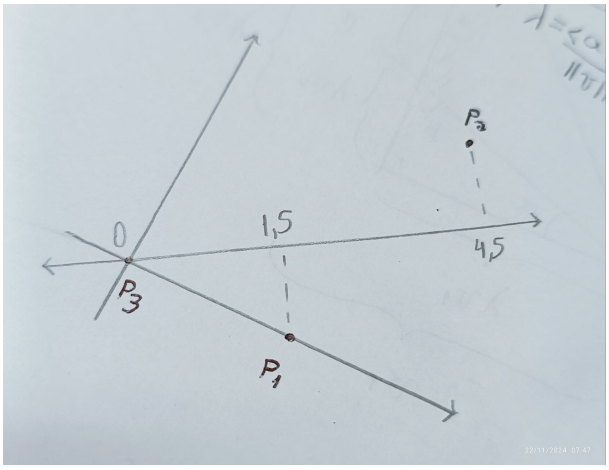

Figura 6.11: LEGENDA.

Sendo assim, conseguimos representar esses pontos a partir da melhor reta possível que os represente e preserve suas distâncias e espaçamentos o mais corretamente possível. Além disso, veja que:

$$\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1.5 & 4.5 & 0\end{bmatrix} = \begin{bmatrix}1.5 & 4.5 & 0\\1.5 & 4.5 & 0\end{bmatrix} \tag{2}$$

Ou seja, por (1) e por (2) temos que:

$$\begin{bmatrix}3 & 4 & 0\\0 & 5 & 0\end{bmatrix} \approx \begin{bmatrix}1.5 & 4.5 & 0\\1.5 & 4.5 & 0\end{bmatrix}$$

Ou seja:

$$\begin{bmatrix}3 & 4 & 0\\0 & 5 & 0\end{bmatrix} \approx \begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1.5 & 4.5 & 0\end{bmatrix}$$

Ou seja, conseguimos achar a melhor aproximação possível de matrizes V e E tais que $VE \approx A^T$

Perceba que estamos realizando uma comparação entre matrizes mas não definimos o que isso significa. Quando comparávamos vetores, fazíamos $\|v - w\|^2$. Com matrizes faremos o mesmo, mas utilizamos a norma matricial de Frobenius tal que, sendo A uma matriz:

$$\|A\|^2 = \sum_{i=1}^{n} \|A_i\|^2$$

Em outras palavras, a norma de uma matriz A é a soma das normas de suas colunas.

Dessa forma, criamos um algoritmo que, com dados de entrada, retorna vetores V e E tais que minimizam $\|A^T - VE\|^2$.

6.4 Outros Componentes Principais

A ideia por trás do segundo componente principal é bem semelhante ao primeiro: Antes estávamos buscando o maior autovalor para, assim, encontrar seu respectivo autovetor. O que faremos nesse instante será algo semelhante, mas dessa vez com o segundo autovalor.

Dessa vez, dado pontos no $\mathbb{R}^3$, gostaríamos de representá-los em um plano gerado por vetores ortogonais z_1 e z_2.

Para a projeção de um ponto p no plano, temos que sua projeção γ pode ser escrita como:

$$\|\gamma^2\| = \left(\frac{p^T z_1}{\|z_1\|^2}\right)^2 + \left(\frac{p^T z_2}{\|z_2\|^2}\right)^2$$

Todo: legenda!

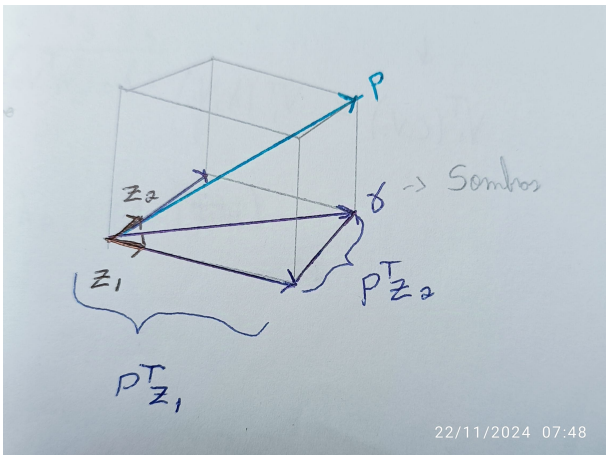

Figura 6.12: LEGENDA.

Veja então que o erro total do sistema com n pontos será:

$$\sum_{i=1}^{n} \left(\frac{p^T z_1}{\|z_1\|^2}\right)^2 + \left(\frac{p^T z_2}{\|z_2\|^2}\right)^2$$

Que é o mesmo que:

$$\|Az_1\|^2 + \|Az_2\|^2$$

Tal que:

$$A = \begin{bmatrix} - & p_1 & - \\ & \vdots & \\ - & p_n & - \end{bmatrix}$$

Sendo assim, gostaríamos de maximizar as projeções e minimizar o erro, assim como fizemos anteriormente.

Podemos calcular o Argmax_{z_2,z_1} de $\|Az_1\|^2 + \|Az_2\|^2$ para encontrar os valores de z_1 e z_2. Ao termos feito esse cálculo para a projeção no $\mathbb{R}^1$, encontramos que o autovetor associado ao maior autovalor era o $\text{Argmax}_v\ h(v)$.

Todo: Seria benéfico provar dessa forma também além de apenas a ideia gulosa

Ideia Gulosa: Encontrar a melhor reta que aproxime os pontos no $\mathbb{R}^1$ e depois buscar a melhor reta ortogonal a esta que aproxima os pontos.

Teorema 6.4.1. *O erro do nosso método que aproxima as linhas de A por um subespaço S de dimensão k no $\mathbb{R}^n$ é minimizado quando S é gerado pelos autovetores associados aos k maiores autovalores de A^TA.*

Demonstração. Analogamente ao feito em 6.2.14, precisamos mostrar que $v_1, \dots v_k$ associados, respectivamente, aos autovalores $\lambda_1, \dots, \lambda_k$ maximizam as sombras (e minimizam o erro) tal que:

$$\frac{\|Av_1\|^2}{\|v_1\|^2} + \cdots + \frac{\|Av_k\|^2}{\|v_k\|^2} \geq \frac{\|Aw_1\|^2}{\|w_1\|^2} + \cdots + \frac{\|Aw_k\|^2}{\|w_k\|^2}$$

Todo: Falta provar que podemos fazer essa escolha de w ortogonais aos ws anteriores

Escolhemos $w_i\ \forall i \in \{1, \dots, n\}$ ortogonais visto que estamos preocupado com o subespaço que eles geram, e não os vetores em si, e que $\langle w_i, w_j \rangle = 0\ \forall i \neq k$. Além disso, temos que $v_i \perp v_j\ \forall i \neq j$ por 6.2.11.

Caso Base: para $n = 1$ temos $\dfrac{\|Av_1\|^2}{\|v_1\|^2} \geq \dfrac{\|Aw_1\|^2}{\|w_1\|^2}$ que provamos em 6.2.14.

Hipótese de Indução: para $n = k$, temos que

$$\frac{\|Av_1\|^2}{\|v_1\|^2} + \cdots + \frac{\|Av_k\|^2}{\|v_k\|^2} \geq \frac{\|Aw_1\|^2}{\|w_1\|^2} + \cdots + \frac{\|Aw_k\|^2}{\|w_k\|^2}$$

Passo de Indução: para $n = k + 1$ temos:

$$\frac{\|Av_1\|^2}{\|v_1\|^2} + \cdots + \frac{\|Av_k\|^2}{\|v_k\|^2} + \frac{\|Av_{k+1}\|^2}{\|v_{k+1}\|^2} \geq \frac{\|Aw_1\|^2}{\|w_1\|^2} + \cdots + \frac{\|Aw_k\|^2}{\|w_k\|^2} + \frac{\|Aw_{k+1}\|^2}{\|w_{k+1}\|^2}$$

Pela nossa hipótese de indução, sabemos que os termos de 1 a k respeitam a desigualdade. Sendo assim, falta provar para o termo restante que, por sua vez, seria:

$$\frac{\|Av_{k+1}\|^2}{\|v_{k+1}\|^2} \geq \frac{\|Aw_{k+1}\|^2}{\|w_{k+1}\|^2}$$

Que, pelo teorema 6.2.8, seria o mesmo que provar que:

$$\lambda_{k+1} \geq \frac{\|Aw_{k+1}\|^2}{\|w_{k+1}\|^2}$$

Pelo teorema 6.2.13, a parcela direta da equação pode ser reescrita como:

$$\lambda_{k+1} \geq \frac{\sum_{i=1}^{n} \alpha_i^2 \lambda_i}{\sum_{i=1}^{n} \alpha_i^2}$$

Mas como w_{k+1} é necessariamente ortogonal a w_i, $\forall i \in \{1, \ldots, k\}$, então temos que w_{k+1} não tem influência de v_i $\forall i \in \{1, \ldots, k\}$. Logo, não tem influência de λ_i. Sendo assim, temos que $\alpha_i = 0$ e que a equação pode ser reescrita como:

$$\lambda_{k+1} \geq \frac{\sum_{i=k+1}^{n} \alpha_i^2 \lambda_i}{\sum_{i=k+1}^{n} \alpha_i^2}$$

E assim, temos que:

$$\begin{aligned}
\lambda_{k+1} &\geq \frac{\sum_{i=k+1}^{n} \alpha_i^2 \lambda_i}{\sum_{i=k+1}^{n} \alpha_i^2} \\
\sum_{i=k+1}^{n} \alpha_i^2 \lambda_{k+1} &\geq \sum_{i=k+1}^{n} \alpha_i^2 \lambda_i \\
\sum_{i=k+1}^{n} \alpha_i^2 \lambda_{k+1} - \sum_{i=k+1}^{n} \alpha_i^2 \lambda_i &\geq 0 \\
\sum_{i=k+1}^{n} \alpha_i^2 (\lambda_{k+1} - \lambda_i) &\geq 0
\end{aligned}$$

Mas como $\lambda_{k+1} \geq \lambda_i$, então temos que o lado esquerdo da equação é necessariamente não negativo e, portanto, obedece a desigualdade.

Logo, provamos que a equação é de fato limitada por cima pelos autovalores de C. Assim, provamos que o erro do nosso método que aproxima as linhas de A por um subespaço S de dimensão k no $\mathbb{R}^n$ é minimizado quando S é gerado pelos autovetores associados aos k maiores autovalores de $A^T A$.

□

6.5 Exemplo de Redução de Dimensionalidade para Duas Dimensões

Dado quatro pontos p_1, p_2, p_3 e p_4 tais que:

$$p_1 = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} \quad p_2 = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \quad p_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad p_4 = \begin{bmatrix} -\sqrt{3} \\ 0 \\ 0 \end{bmatrix}$$

Queremos encontrar a melhor aproximação para esses pontos em um plano (no $\mathbb{R}^2$). O teorema 6.4.1 nos garante que as duas retas ortogonais que formarão o plano que minimizam o erro são aquelas que tem como vetor diretor os dois autovetores associados aos maiores autovalores da matriz C formada por $A^T A$ tal que:

$$\begin{aligned} C &= A^T A \\ C &= \begin{bmatrix} -1 & 0 & 2 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \\ -\sqrt{3} & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & -\sqrt{3} \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix} \\ C &= \begin{bmatrix} 4 & 0 & -2 \\ 0 & 4 & 0 \\ -2 & 0 & 4 \end{bmatrix} \end{aligned}$$

Visto que temos C, precisamos calcular seus autovalores e autovetores:

$$\begin{aligned} \det \begin{bmatrix} 4 & 0 & -2 \\ 0 & 4 & 0 \\ -2 & 0 & 4 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} &= 0 \\ \det \begin{bmatrix} 4 & 0 & -2 \\ 0 & 4 & 0 \\ -2 & 0 & 4 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} &= 0 \\ \det \begin{bmatrix} 4-\lambda & 0 & -2 \\ 0 & 4-\lambda & 0 \\ -2 & 0 & 4-\lambda \end{bmatrix} &= 0 \\ (4-\lambda)^3 - 4(4-\lambda) &= 0 \\ (4-\lambda)^3 - 4(4-\lambda) &= 0 \\ (4-\lambda)(16 - 8\lambda + \lambda^2 - 4) &= 0 \\ (\lambda - 4)(\lambda - 6)(\lambda - 2) &= 0 \end{aligned}$$

Ou seja, encontramos que os autovalores são $\lambda_1 = 6$, $\lambda_2 = 4$ e $\lambda_3 = 2$. Visto que queremos representar nossos pontos em um plano no $\mathbb{R}^2$, precisamos escolher os dois maiores autovalores para calcular o PCA_1 e o PCA_2 (autovetores associados aos autovalores λ_1 e λ_2).

Cálculo do PCA_1 utilizando $\lambda_1 = 6$:

Todo: Aqui temos o mesmo problema da seção 6.3, em que utilizamos essa fórmula para calcular o autovetor mas sem fonte satisfatória

$$\begin{bmatrix} 4-6 & 0 & -2 \\ 0 & 4-6 & 0 \\ -2 & 0 & 4-6 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

ao resolver o sistema, temos que $w_2 = 0$ e que $w_1 = -w_3$. Sendo assim, temos que o autovetor associado ao maior autovalor é:

$$PCA1 = w_3 \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Cálculo do PCA_2 utilizando $\lambda_2 = 4$:

$$\begin{bmatrix} 4-4 & 0 & -2 \\ 0 & 4-4 & 0 \\ -2 & 0 & 4-4 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

ao resolver o sistema, temos que $v_1 = v_3 = 0$. Sendo assim, temos que o autovetor associado ao segundo maior autovalor é:

$$PCA_2 = v_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Seja $w = PCA_1$ e $v = PCA_2$.

Veja que, de fato, w e v são ortogonais:

$$w^T v = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 0 + 0 + 0 = 0$$

Sendo assim, eles formam uma base ortogonal:

Todo: legenda!

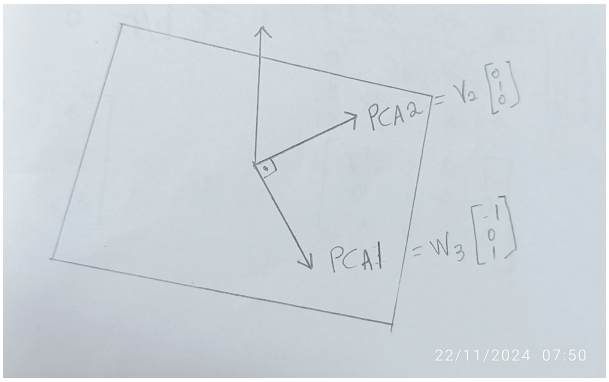

Figura 6.13: LEGENDA.

Agora precisamos projetar os pontos de entrada do problema no plano gerado por w e v. Veja que, como é uma projeção, podemos projetar em w e v ou em w e v normalizados que o resultado será o mesmo.

Visto que w e v são ortogonais, temos que:

$$\text{proj}_{plano}(p_i) = \text{proj}_{\overline{w}}(p_i) + \text{proj}_{\overline{v}}(p_i)$$

Todo: legenda!

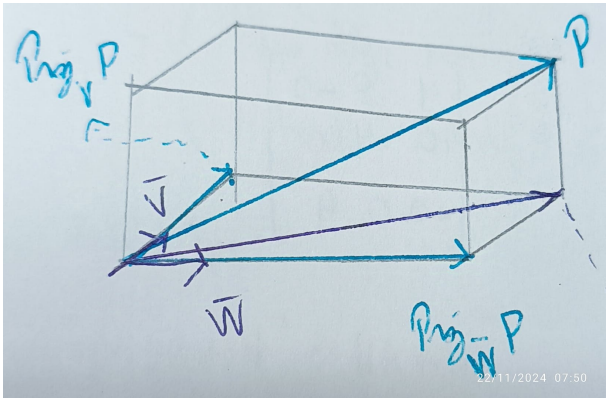

Figura 6.14: LEGENDA.

Visto que a fórmula da projeção é $\text{proj}_z(a) = \dfrac{\langle z, a\rangle}{\|z\|^2} z$, temos que, se z é normalizado, então: $\dfrac{\langle z, a\rangle}{\|z\|^2} z = \langle z, a\rangle\, z$. Sendo assim, temos:

$$\overline{w} = \frac{w}{\|w\|} \quad \overline{v} = \frac{v}{\|v\|}$$

Calculando as projeções em $\overline{w}$:

$$\text{proj}_{\overline{w}}(p_1) = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} \overline{w} = \frac{3}{\sqrt{2}} \overline{w}$$

$$\text{proj}_{\overline{w}}(p_2) = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \overline{w} = 0\overline{w}$$

$$\text{proj}_{\overline{w}}(p_3) = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \overline{w} = 0\overline{w}$$

$$\text{proj}_{\overline{w}}(p_4) = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} -\sqrt{3} \\ 0 \\ 0 \end{bmatrix} \overline{w} = \frac{\sqrt{3}}{\sqrt{2}} \overline{w}$$

Calculando as projeções em $\overline{v}$:

$$\text{proj}_{\overline{v}}(p_1) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} \overline{v} = 0\overline{v}$$

$$\text{proj}_{\overline{v}}(p_2) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \overline{v} = 2\overline{v}$$

$$\text{proj}_{\overline{v}}(p_3) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \overline{v} = 0\overline{v}$$

$$\text{proj}_{\overline{v}}(p_4) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} -\sqrt{3} \\ 0 \\ 0 \end{bmatrix} \overline{v} = 0\overline{v}$$

Ao sabermos que $\text{proj}_{plano}(p_i) = \text{proj}_{\overline{w}}(p_i) + \text{proj}_{\overline{v}}(p_i)$, temos então que:

$$\text{proj}_{plano}(p_1) = \text{proj}_{\overline{w}}(p_1) + \text{proj}_{\overline{v}}(p_1) = \frac{3}{\sqrt{2}} \overline{w} + 0\overline{v}$$

$$\text{proj}_{plano}(p_2) = \text{proj}_{\overline{w}}(p_2) + \text{proj}_{\overline{v}}(p_2) = 0\overline{w} + 2\overline{v}$$

$$\text{proj}_{plano}(p_3) = \text{proj}_{\overline{w}}(p_3) + \text{proj}_{\overline{v}}(p_3) = 0\overline{w} + 0\overline{v}$$

$$\text{proj}_{plano}(p_4) = \text{proj}_{\overline{w}}(p_4) + \text{proj}_{\overline{v}}(p_4) = \frac{\sqrt{3}}{\sqrt{2}}\overline{w} + 0\overline{v}$$

Todo: legendas

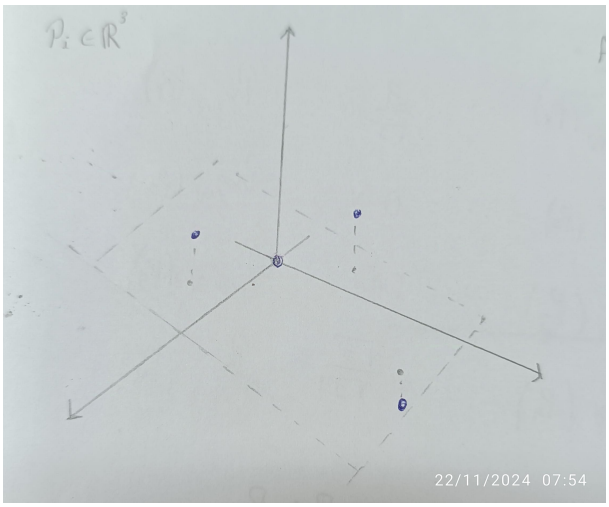

Figura 6.15: LEGENDA.

Ou seja, temos que:

$$\begin{bmatrix} | & | \\ \overline{w} & \overline{v} \\ | & | \end{bmatrix} \begin{bmatrix} \frac{3}{\sqrt{2}} & 0 & 0 & \frac{\sqrt{3}}{\sqrt{2}} \\ 0 & 2 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \\ \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} \frac{3}{\sqrt{2}} & 0 & 0 & \frac{\sqrt{3}}{\sqrt{2}} \\ 0 & 2 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -3 & 0 & 0 & -\frac{\sqrt{3}}{2} \\ 0 & 2 & 0 & 0 \\ \frac{3}{2} & 0 & 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$$

Visto que nossa matriz A^T inicial é:

$$A^T = \begin{bmatrix} -1 & 0 & 0 & -\sqrt{3} \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

Temos que:

$$\begin{bmatrix} -3 & 0 & 0 & -\frac{\sqrt{3}}{2} \\ 0 & 2 & 0 & 0 \\ \frac{3}{2} & 0 & 0 & \frac{\sqrt{3}}{2} \end{bmatrix} \approx \begin{bmatrix} -1 & 0 & 0 & -\sqrt{3} \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

Veja que, caso tivéssemos utilizado apenas o PCA_1 teríamos:

$$\begin{bmatrix} -1 & 0 & 0 & -\sqrt{3} \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix} \approx \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{3}{\sqrt{2}} & 0 & 0 & \frac{\sqrt{3}}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} -\frac{3}{2} & 0 & 0 & -\frac{\sqrt{3}}{2} \\ 0 & 0 & 0 & 0 \\ \frac{3}{2} & 0 & 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$$

Ou seja, utilizando apenas o PCA_1, temos que:

$$\begin{bmatrix} -1 & 0 & 0 & -\sqrt{3} \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix} \approx \begin{bmatrix} -\frac{3}{2} & 0 & 0 & -\frac{\sqrt{3}}{2} \\ 0 & 0 & 0 & 0 \\ \frac{3}{2} & 0 & 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$$

E utilizando o PCA_1 e o PCA_2 temos que:

$$\begin{bmatrix} -1 & 0 & 0 & -\sqrt{3} \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix} \approx \begin{bmatrix} -3 & 0 & 0 & -\frac{\sqrt{3}}{2} \\ 0 & 2 & 0 & 0 \\ \frac{3}{2} & 0 & 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$$

Cada PCA_i está associado ao $autovalor_i$. No caso dos dados de entrada terem dimensão n, temos que:

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \frac{\|Av_i\|^2}{\|v_i\|^2}$$

Visto que $\frac{\|Av_i\|^2}{\|v_i\|^2}$ é o tamanho das projeções (sombras), sabemos que no caso $\mathbb{R}^n$, a projeção dos pontos será exatamente eles mesmos. Logo:

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \frac{\|Av_i\|^2}{\|v_i\|^2} = \left\|A^T\right\|^2$$

No caso do exemplo acima, temos que $\lambda_1 + \lambda_2 + \lambda_3 = 6 + 4 + 2 = 12$. Sendo assim, temos que λ_1 sozinho representa $\frac{6}{12}$ dos dados, enquanto λ_2 representa $\frac{4}{12}$ e λ_3 representa $\frac{2}{12}$.

Sendo assim, ao utilizar apenas o PCA_1 representamos $\frac{6}{12} = 50\%$ dos dados, enquanto com o PCA_1 e PCA_2 representamos $\frac{6+4}{12} \approx 83.3\%$ dos dados e com PCA_1, PCA_2 e PCA_3 representamos $\frac{6+4+2}{12} = 100\%$ dos dados.

Todo: legenda

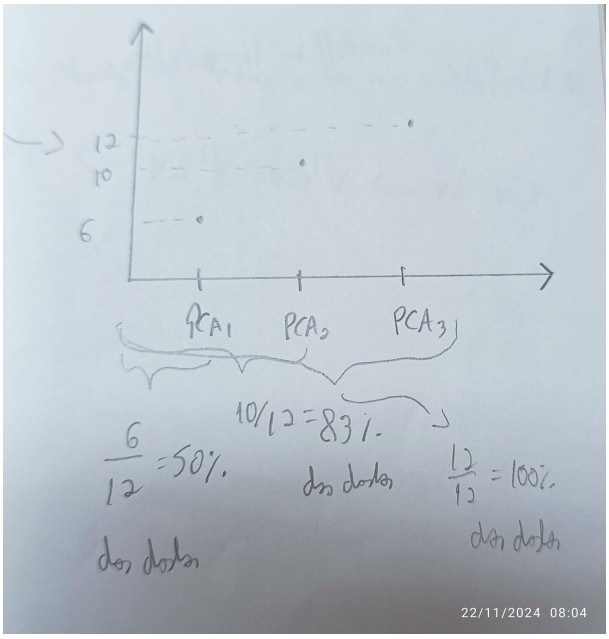

Figura 6.16: LEGENDA.

Dado um outro conjunto de dados de exemplo, temos que:

Todo: legenda

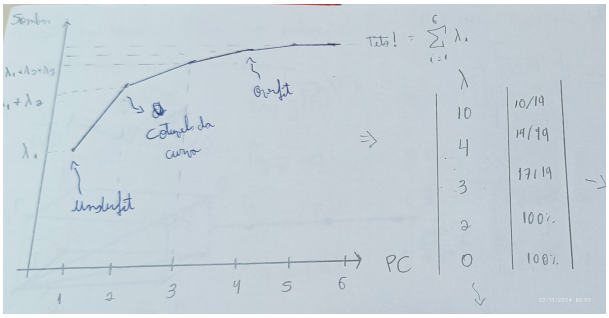

Figura 6.17: LEGENDA.

No gráfico acima temos que:

Autovalores	Representação aproximada	Percentual aproximado de representação
10	10/19	52.6%
4	14/19	74.6%
3	17/19	89.5%
2	19/19	100 %
0.001	19/19	100 %

Sendo assim, gostaríamos muito de dizer que a matriz dos dados de entrada é de posto 4 visto que ela é praticamente 100% representada pelos 4 maiores autovalores e considerar o autovalor 0.001 como um erro de máquina.

6.6 Decomposição em Valores Singulares

em 6.4 vimos que é possível fatorarmos uma matriz A em um produto de matrizes menores tais que uma delas seja formada por alguns dos maiores autovetores de A e a outra seja formada pelos endereços que representam as projeções dos vetores de A em seus autovetores. Nessa seção encontraremos um significado diferente para a segunda matriz da decomposição e mostraremos como fatorar A dessa vez como o produto de 3 matrizes.

Teorema 6.6.1. *A norma ao quadrado de Frobenius de uma matriz A é igual a norma ao quadrado de Frobenius da mesma matriz mas transposta. Ou seja,* $\|A\|^2 = \left\|A^T\right\|^2$

Demonstração.

$$\begin{aligned}\|A\|^2 &= \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}^2 \\ &= \sum_{j=1}^{m}\sum_{i=1}^{n} (a_{ji}^2)^2 \\ &= \left\|A^T\right\|^2\end{aligned}$$

Sendo assim, mostramos que $\|A\|^2 = \left\|A^T\right\|^2$. □

Teorema 6.6.2. *Se VE é a melhor fatoração possível para A, então E^TV^T é a melhor fatoração possível para A^T. Sendo a melhor fatoração possível aquela que minimiza o erro na norma de Frobenius.*

Demonstração. Visto que VE é a melhor fatoração possível para A. Ou seja, $A \approx VE$. Basta provar que ao fazer $(A)^T \approx (VE)^T \implies A^T \approx E^TV^T$ o erro será mantido.

Mostramos em 6.6.1 que para uma matriz qualquer, a sua norma ao quadrado de Frobenius é igual a norma ao quadrado de Frobenius de sua transposta. Sendo assim, sabemos que: $\|A\|^2 = \left\|A^T\right\|^2 \implies \|VE\|^2 = \left\|E^TV^T\right\|$.

Sendo assim, provamos que se VE é a melhor aproximação possível para A, então a melhor aproximação possível para A^T é E^TV^T. □

Visto então que $A \approx VE$ e $A^T \approx E^TV^T$, e sabemos que as matrizes que melhor aproximam uma matriz são o produto de seus maiores autovetores e da projeção de suas colunas nesses autovetores, então podemos concluir que E^T é a matriz que tem como colunas os autovetores da matriz ${A^T}^TA^T$. Dessa forma, podemos concluir que E é a matriz que tem como linhas autovetores de ${A^T}^TA^T$. Além disso, provamos em 6.2.11 que esses autovetores são ortogonais. Sendo assim, temos então que a decomposição A em VE tem V como matriz ortogonal (pois é formada pelos autovetores de A^TA) e E também matriz com vetores em linha ortogonais entre si.

Veja que $A_{n\times m}$ por ser fatorado de forma exata por um produto de matrizes $V_{n\times k}$ e $E_{k\times m}$ tais que $A_{n\times m} = V_{n\times k}E_{k\times m}$ e V é matriz ortonormal, temos que:

$$A = \begin{bmatrix} | & | & | \\ u_1 & \dots & u_m \\ | & | & | \end{bmatrix} \begin{bmatrix} — & v_1^T & — \\ — & \vdots & — \\ — & v_k^T & — \end{bmatrix}$$

tal que u_i, $i \in \{1, \dots, m\}$, é normal e v_j^T, $j \in \{1, \dots, k\}$, não necessariamente. Mas veja que podemos dividir cada um deles pela sua norma para normalizarmos. Em outras palavras, sabemos que:

$$\left\| v_i^T \right\| \hat{v}_i^T = v_i^T$$

Logo:

$$\begin{bmatrix} — & v_1^T & — \\ — & \vdots & — \\ — & v_k^T & — \end{bmatrix} = \begin{bmatrix} \left\| v_1^T \right\| & 0 & \ddots \\ 0 & \ddots & 0 \\ \ddots & 0 & \left\| v_k^T \right\| \end{bmatrix} \begin{bmatrix} — & \hat{v}_1^T & — \\ — & \vdots & — \\ — & \hat{v}_k^T & — \end{bmatrix}$$

Dessa forma, podemos escrever A como:

$$A = \begin{bmatrix} | & | & | \\ u_1 & \dots & u_m \\ | & | & | \end{bmatrix} \begin{bmatrix} \left\| v_1^T \right\| & 0 & \ddots \\ 0 & \ddots & 0 \\ \ddots & 0 & \left\| v_k^T \right\| \end{bmatrix} \begin{bmatrix} — & \hat{v}_1^T & — \\ — & \vdots & — \\ — & \hat{v}_k^T & — \end{bmatrix}$$

Ou então, de forma sucinta, $A = UDV^T$, tal que U e V^T sejam matrizes ortogonais e D seja uma matriz diagonal.

Todo: Os valores singulares são necessariamente não negativos por serem normas, mas eles nos dizem algo sobre os autovalores da matriz?

Todo: Acho que faltou mostrar que o erro na norma de Frobenius para o PCA terá uma relação com os autovalores?

Capítulo 7

Outros assuntos

7.1 Regressão Logística

Seja Ω um conjunto de informações tais que Ω é bi-particionado em 2 subconjuntos A e B. Dado uma nova informação, gostaríamos de estimar em qual subconjunto é mais provável que ela esteja.

Conforme visto em 6.4, podemos reduzir a dimensão desses dados para o $\mathbb{R}^1$, por exemplo, e adicionar uma segunda coordenada que serve como uma espécie de indicativo de qual subconjunto aquela informação pertence. Por exemplo, podemos colocar como segunda coordenada dos elementos de A o valor 0 e de B o valor 1 para que, dessa forma, possamos, por meio de uma regressão, encontrar uma curva que aproxime os pontos.

Todo: Legenda! Imagem 1

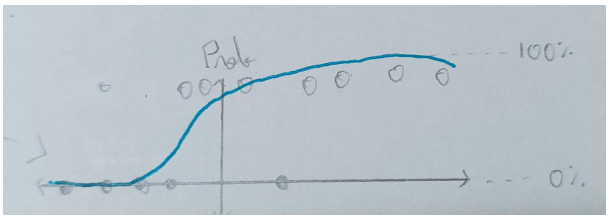

Figura 7.1: LEGENDA.

Veja que uma função capaz de aproximar pontos dividimos em alturas 0 e 1 é $p(x) = \frac{e^{c_0+c_1x}}{1+e^{c_0+c_1x}}$ visto que:

$$\lim_{x\to-\infty} \frac{e^{c_0+c_1x}}{1+e^{c_0+c_1x}} = \begin{cases} 1 & \text{se } c_1 < 0 \\ \frac{e^{c_0}}{1+e^{c_0}} & \text{se } c_1 = 0 \\ 0 & \text{se } c_1 > 0 \end{cases}$$

$$\lim_{x\to\infty} \frac{e^{c_0+c_1x}}{1+e^{c_0+c_1x}} = \begin{cases} 0 & \text{se } c_1 < 0 \\ \dfrac{e^{c_0}}{1+e^{c_0}} & \text{se } c_1 = 0 \\ 1 & \text{se } c_1 > 0 \end{cases}$$

Indiferentemente de A apresentar segunda coordenada 0 ou 1, por estarmos utilizando uma regressão, os valores de c_0 e c_1 se ajustarão conforme o formato da função para aproximar os pontos de A e B. Sendo assim, não precisamos nos preocupar com restrição no valor de c_1.

Vimos em 1.2 que o erro dos mínimos quadrados é:

$$E(cs) = \|\mathcal{A}cs - ys\|^2 = (f(x_1) - y_1)^2 + \cdots + (f(x_i) - y_i)^2 + \cdots + (f(x_n) - y_n)^2$$

Sendo n a quantidade de pontos e $i \in \{1, \ldots, n\}$.

Sendo assim, gostaríamos de encontrar os valores de c_0 e c_1 tais que o erro da regressão seja minimizado.

Em outras palavras, queremos calcular:

$$\text{Min}_{c_0,c_1}\ (p(x_1) - (1 \text{ ou } 0))^2 + \cdots + (p(x_i) - (1 \text{ ou } 0)) + \cdots + (p(x_n) - (1 \text{ ou } 0))$$

Tal que (1 ou 0) seja escolhido de acordo com a segunda coordenada que atribuímos ao ponto x_i.

O problema desse cálculo é, que por mais que ele seja exatamente o que gostaríamos de fazer, a manipulação algébrica até chegar no resultado desejado pode não fazer sentido visto que ao derivar a equação e igualar a 0 teremos uma derivada não linear por conta da função $p(x)$ e também não parece fazer muito sentido o cálculo da probabilidade ao quadrado. Sendo assim, faremos uma estimativa para esse procedimento por máxima verossimilhança.

Queremos que os valores de alguns $p(x_i)$ sejam altos a depender de qual conjunto atribuímos os valores de 0 e 1. Sendo assim, gostaríamos de calcular:

$$\text{Argmax}_{c_0,c_1} \prod_{y_i=1} p(x_i) \cdot \prod_{y_i=0} (1 - p(x_i))$$

Todo: Fazer essa demonstração

Teorema 7.1.1. *$Argmax_x\ f(x) = Argmax_x\ \log(f(x))$*

Demonstração. □

Sabemos por 7.1.1 que:

$$\begin{aligned}\text{Argmax}_{c_0,c_1} \prod_{y_i=1} p(x_i) \cdot \prod_{y_i=0} (1-p(x_i)) &= \text{Argmax}_{c_0,c_1} \log\left(\prod_{y_i=1} p(x_i) \cdot \prod_{y_i=0} (1-p(x_i))\right) \\ &= \text{Argmax}_{c_0,c_1} \sum_{y_i=1} \log(p(x_i)) + \sum_{y_i=0} \log(1-p(x_i))\end{aligned}$$

Derivando em c_1 e igualando a 0:

$$\begin{aligned}\frac{\partial}{\partial c_1}\text{Argmax}_{c_0,c_1} \sum_{y_i=1} \log(p(x_i)) + \sum_{y_i=0} \log(1-p(x_i)) = 0 \\ \sum_{i=0}^{n} \frac{1}{p(x_i)} p_{c_1}(x_i)x_i - \sum_{i=0}^{n} \frac{1}{1-p(x_i)} p_{c_i}(x_i)x_i = 0\end{aligned}$$

Mas veja que $p_{c_1}(x_i)$ é:

$$\begin{aligned}p_{c_1}(x) &= \left(\frac{e^{c_0+c_1x}}{1+e^{c_0+c_1x}}\right)_{c_i} \\ &= \frac{e^{c_0+c_1x}(1+e^{c_0+c_1x}) - e^{c_0+c_1x}e^{c_0+c_1x}}{(e^{c_0+c_1x})^2} \\ &= \frac{e^{c_0+c_1x}(1+e^{c_0+c_1x} - e^{c_0+c_1x})}{(1+e^{c_0+c_1x})^2} \\ &= \frac{e^{c_0+c_1x}}{(1+e^{c_0+c_1x})^2} \\ &= \frac{e^{c_0+c_1x}}{(1+e^{c_0+c_1x})} \frac{1}{(1+e^{c_0+c_1x})} \\ &= p(x_i)(1-p(x_i))\end{aligned}$$

Logo:

$$\sum_{y_i=1} \frac{1}{p(x_i)} p(x_i)(1-p(x_i))x_i - \sum_{y_i=0} \frac{1}{1-p(x_i)} p(x_i)(1-p(x_i))x_i = 0$$

$$\sum_{y_i=1} \frac{1}{\cancel{p(x_i)}} \cancel{p(x_i)}(1-p(x_i))x_i - \sum_{y_i=0} \frac{1}{\cancel{1-p(x_i)}} p(x_i)\cancel{(1-p(x_i))}x_i = 0$$

$$\sum_{y_i=1} (1-p(x_i))x_i - \sum_{y_i=0} p(x_i)x_i = 0$$

$$\sum_{i=0}^{n} y_i(1-p(x_i))x_i - \sum_{i=0}^{n} (1-y_i)p(x_i)x_i = 0$$

$$\sum_{i=0}^{n} y_i(1-p(x_i))x_i - (1-y_i)p(x_i)x_i = 0$$

$$\sum_{i=0}^{n} (y_i - y_i p(x_i) - p(x_i) + y_i p(x_i))x_i = 0$$

$$\sum_{i=0}^{n} (y_i - p(x_i))x_i = 0$$

Bibliografia

[1] .
[2] .