UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71956
TAINARA MARIA WALTER SCUISS(71956);
FELIPE BENDER LAVALLE (71725);
BRYAN MARTINS PEREIRA (71637);
221EC102 - 1 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x,y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC01.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71956 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71806
GUSTAVO FELIX GONCALVES PIN(71806);
VINICIUS TIAGO FONTANELLA (71987);
PAULO HENRIK VALERIO DOS AN(71925);
221EC102 - 2 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC02.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71806 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71606
ALEXANDRE POLSWUT (71606);
GABRIELA BIANCHI TODT (71756);
CAUA DE PADUA XAVIER (71644);
221EC102 - 3 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC03.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71606 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71699
EMELYN SURECK (71699);
GUILHERME CONDE DIAS (71787);
ABNER FELIPE NOGUEIRA (71594);
221EC102 - 4 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x,y)\delta(x\sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC04.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71699 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71763
GABRIELA CARLONE TSCHOEKE (71763);
HIROSHI MISAKI (71820);
EMILY DEBORA RIBEIRO DE MAT(71701);
221EC102 - 5 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC05.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71763 - li/it

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC06.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71675
DANIEL DREHMER DE MELLO E S(71675);
GABRIELE TOMAZONI DINIZ (71770);
JOAO VICTOR LINCOLN CUSTODI(71868);
221EC102 - 7 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC07.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71918
MARLI MORO (71918);
RAFAEL DIAS CORREA (71932);
LUCAS GRABOWSKI TREVISAN (71899);
221EC102 - 8 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC08.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71718
ENDREU ROBERTO DE JESUS VEL(71718);
FERNANDA GIACOMIN GEMELI (71749);
EDUARDA VIEIRA SAGGIN (71682);
221EC102 - 9 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x,y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC09.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71718 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71882
KETYLLI JOJEN WU (71882);
STELLA DE SOUZA STOJCZAN (71949);
VICTOR HUGO BARROZO MARINON(71970);
221EC102 - 10 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC10.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71882 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71844
ISABELY PAOLA LEAL DA SILVA(71844);
ARTHUR BENECK BORDIGNON (71620);
FELIPE CONTE PESSOA DE OLIV(71732);
221EC102 - 11 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC11.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71844 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:56:06.8
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71613
ANGELUZ KAUE FROSSARD PAIVA(71613); TIAGO VENSON (71963);
GUILHERME HENRIQUE GREGORIN(71794);
221EC102 - 12 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC12.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71613 - li/it

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades" e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC13.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31EC14.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:48:31.4
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71394
FRANCIANE GUIMARAES LOPES (71394);
CARLOS EDUARDO BRAGA FAVERO(71282);
FELIPE GAIEVSKI ALMEIDA (71349);
221FI102 - 1 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI01.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71394 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:48:31.4
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71325
ERICK BITTAR BELTRAMI (71325);
GIOVANNA MIRANDA MENDES (71420);
FILIPE SANTINI ROCHA (71387);
221FI102 - 2 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI02.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71325 - li/it

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades" e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI03.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:48:31.4
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71299
DOUGLAS NAPOLEAO SERMANN (71299); EWERTON GUSTAVO DE SOUZA (71332); GUSTAVO DE BRITO GOMES (72159);
221FI102 - 4 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI04.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71299 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:48:31.4
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71318
EDUARDO ANTONIO SOMER (71318);
PAULA TAYNAH SANTOS PEREIRA(71556);
MATHEUS RODRIGUES DE JESUS (71532);
221FI102 - 5 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x,y)\delta(x\sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI05.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:48:31.4
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71370
FERNANDO FERNANDES FIGUEIRE(71370);
LUCAS MASSAYUKI GOHARA (71501);
LEANDRO CESAR GUIMARAES (71482);
221FI102 - 6 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI06.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.

Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm

GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71370 - li/it

Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71570
VICTOR DOS SANTOS SANTIN (71570);
ARTHUR FRANCISCO ARAUJO BRU(71268);
FERNANDO BATISTA JUNIOR (71363);
221FI102 - 7 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI07.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:48:31.4
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71499
LEONARDO AMORIM CARPWISKI (71499);
FRANCO LUIZ FONTOURA TEIXEI(71406);
GUSTAVO DE LIMA VIEIRA (71444);
221FI102 - 8 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x,y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI08.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71499 - li/it

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:48:31.4
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71413
GABRIEL FRANCISCO NASCIMENT(71413);
PEDRO LUCAS SPENGLER DOS SA(71563);
KAIO VIANNA HETMANEK (71475);
221FI102 - 9 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI09.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:48:31.4
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 71451
JOAO PEDRO DISSENHA MACHADO(71451); VICTOR TARABOLA CORTIANO (71587); LUCAS MELGES FRANCO (71518);
221FI102 - 10 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI10.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-71451 - li/it

JOAO VITOR DE JESUS ANDRADE(71468); ALYSSON FELIPE SILVA (71251); EDUARDA FIGUEIREDO DE OLIVE(71301);
221FI102 - 11 data limite: 20/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y \cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI11.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31FI12.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho `a` será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:54:38.5
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 72016
ARNALDO AUGUSTO LOPES DA SI(72016);
SILAS ANTUNES RODRIGUES FIL(72104);
THAISLAINE RAFAELA DOS REIS(72111);
221ES102 - 1 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31ES01.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:54:38.5
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 72078
LUIZA HELENA DE PAULA (72078);
DAIANE NUNES DOS SANTOS (72023);
MICHELE VARPECHOWSKI CASTEL(72085);
221ES102 - 2 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31ES02.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

102-72078 - li/it

Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 72135
VITOR YUJI KIEMO (72135);
LUIZ FELIPE DOS ANJOS COSTA(72061);
FELIPE TURATTI (72047);
221ES102 - 3 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31ES03.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:
DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 72054
GABRIEL JOSE FERRARINE (72054);
VITOR WALLACE SWISTALSKI (72128);
ANDREA PEREIRA LOBO TAVARES(72009);
221ES102 - 4 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31ES04.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.

UFPR - UP - UTFPR - PUCPr – 14/03/2022 - 20:54:38.5
Prof Dr P Kantek (pkantek@gmail.com)
Sistemas Lineares na prática: Tomografia Computadorizada VIVOk37a, V: 1.05 72030
DOUGLAS DOS SANTOS MAGALHAE(72030);
ANDREA ALBUQUERQUE ALVES (71994);
ROGER ROBERTO DA SILVA (72092);
221ES102 - 5 data limite: 19/04/22 ______ / / ______

Sistemas Lineares na prática: Tomografia Computadorizada

Para encerrar a discussão de sistemas lineares com uma aplicação robusta e real, vai-se trabalhar com a reconstituição de imagens que é feita pela máquina de TCA (tomografia computadorizada axial).

Começa-se a discussão pela presença muito comum de um sistema de m equações a n incógnitas estudando-se agora o que acontece com os diversos valores de m e n.

Quando $m = n$, tem-se o caso estudado até aqui nas aulas anteriores, já que se as m equações são linearmente independentes (nenhuma é múltipla de outra ou combinação linear de outras) o sistema tem uma única solução.

Se $m < n$ o sistema só admite solução se forem arbitrados valores **ad-hoc** para $m - n$ incógnitas.

O caso que vai ser estudado agora é $m > n$ que é o caso mais comum na realidade. Em uma rápida análise, comecemos com um sistema $m = n$ e acrescentemos ao sistema uma nova equação linearmente independente.

Para simplificar usando um caso real suponhamos 3 equações e 3 incógnitas: o sistema tem uma solução x_1, x_2 e x_3. Ao acrescentar a quarta equação o sistema deixa de ter solução exata. Mas, se pegarmos 3 dessas 4 equações (por exemplo as equações 1, 2 e 4) ele voltará a ter solução. Vamos chamá-la de x_1, x_2 e x_4. Se pegarmos as equações 1, 3 e 4, nova solução. Ao resolver o sistema com as 4 equações quer-se o ponto do espaço que minimiza a distância (mínimos quadrados) das soluções de 4 equações tomadas 3 a 3: a saber 1,2,3; 1,2,4; 1,3,4 e 2,3,4.

Esta estratégia faz todo o sentido, já que na vida real não existem soluções exatas, por diversas razões, sendo a principal a omnipresença do ruído, aqui entendido como erro (de leitura, de processamento, de transcriçao, do universo contra mim...).

Não vamos trabalhar com este algoritmo ele é bastante sofisticado, mas vão-se usar os pacotes à nossa disposição (APL, Freemat, Matlab, Python, ...). Neste último não se usa `numpy.linalg.solve` já que este exige matrizes quadradas, mas `numpy.linalg.lstsq`. Este exercício pede que você resolva um sistema com 170 equações a 64 incógnitas.

Tomografia Computadorizada Axial

Trata-se de uma técnica que usa um computador e uma máquina de Raio X para reconstituir imagens de maneira transversal de um corpo. Originalmente na pesquisa do seu inventor, o programador inglês Godfrey Hounsfield, era apenas o cérebro humano já que este – por estar dentro de um capacete de matéria óssea: o crânio – é impermeável ao Raio-X exceto nos casos de rachaduras ou quebra. Mais tarde a técnica deu tão certo que acabou sendo adaptada a outras partes do corpo, mesmo de animais e também em alfândegas, presídios e similares para examinar o interior de qualquer coisa. A propósito até onde sei, o Hounsfield foi o único não médico a ganhar o Nobel de Medicina até hoje.

Feixes de raio X são disparados sobre o corpo em análise e ao invés de serem registrados em fotografias são entregues a um computador (a você...) para processamento. A máquina toda gira ao redor do corpo tirando muitos tomogramas do mesmo objeto. A variação angular é que vai permitir a reconstituição posterior.

O coeficiente de atenuação linear média μ_t de cada pixel é comparado com o coeficiente da água, μ_a, definindo o número CT:

$CT = 1000(\mu_t - \mu_a)/\mu_a$.

A água é utilizada como referência porque seu coeficiente de atenuação é similar ao dos tecidos moles, e é um material fácil de obter para calibrar os aparelhos. O coeficiente 1000 é utilizado para obter números inteiros.

O número CT, ou coeficiente de Hounsfield, é definido como -1000 para o ar e 0 para a água.

A radiação observada, I, está relacionada com a radiação na fonte, I_0, por: $I = I_0 e^{-\mu x}$.

Tecido	CT	Tecido	CT
Ar	-1000	Sangue	35:55
Pulmão	-900:-400	Coágulo	80
Gordura	-110:-65	Músculo	40:60
Água	0	Fígado	50:85
Rim	30	Osso	130:250

Por convenção, altos valores de CT são imageados como branco, e baixos como preto. Como o olho humano não pode distinguir os milhares de coeficientes, utilizamos a técnica de janelas (windowing), para graficar somente os valores em uma certa faixa.

Aquisição de dados

Na tomografia computadorizada mais comum, um tubo com um feixe de cerca de 0,6 mm de diâmetro gira em torno do paciente, e emerge do paciente sobre um detector com aproximadamente 700 sensores, que convertem a intensidade em uma corrente. Cada pulso de raio-X dura 2 a 3 ms, completando uma volta em cerca de 1 s. Cada 360° gera 300 somas.

Cada vez que o tubo emite um pulso, cada detector mede o logaritmo da intensidade que recebe. Este valor representa a soma de todos os números CT dos voxeis atravessados pelo raio, completando uma projeção. Cada voxel é atravessado pelo feixe em diferentes direções, durante a rotação do anel. O número CT de cada voxel está portanto representado em várias somas.

Algoritmo de reconstrução

Vamos pensar em uma sessão transversal de um corpo humano. Imagine também um conjunto de emissores de raio-X, colocados paralelos em uma estrutura rígida que gira em um eixo estabelecido no centro do corpo humano. No lado oposto (ultrapassado o corpo) estão os detectores do raio-X.

À medida em que a máquina gira, inumeros conjuntos de Raios-X vão sendo tomados. Para uma determinada posição angular (digamos horizontal) tem-se um conjunto de x valores de atenuação. O valor x é a quantidade de emissores e detetores, e a atenuação é a diminuição da potência do Raio-X emitido numa ponta e recebido na outra após passar pelas estruturas do corpo que está sendo estudado.

Por exemplo, na horizontal, os raios inferiores, tem que atravessar a coluna (supondo que o paciente esteja deitado de costas), ao passo que os raios superiores ou não atravessam nada, ou apenas alguma gordura (isso para os mais gordinhos). Quando a estrutura estiver tirando raios X na vertical, a coluna do paciente atenuará os raios do meio do conjunto.

Os detectores não tiram fotografias, ao invés eles mandam sinais elétricos proporcionais a potência do Raio-X recebido diretamente para um computador. Este, de posse da informação referente ao ângulo de tomada e dos valores de todos os sensores, vai reconstituindo do corpo que está sendo "cortado".

A técnica mais simples de reconstrução é a chamada "back projection", e todas as demais que vieram depois são melhoramentos e variações desta.

Sua formulação matemática é bem compacta:

$$P(\theta, t) = \int_R f(x, y)\delta(x \sin\theta - y\cos\theta - t)dxdy$$

Essa fórmula pode ser lida como: P é a atenuação medida no sensor t, posicionado em um ângulo θ em relação ao eixo horizontal. x, y são as coordenadas dos pontos da imagem e δ é um filtro que tem a função de só considerar os pontos que estão no caminho do fluxo t de Raios X.

☞ Para você fazer

Neste exercício imagine uma tomografia de um único plano, de um corpo quadrado de 8×8 binário (valendo 0 ou 1 apenas). A cada tomograma há 10 feixes, e depois há um giro de 5^o graus no sentido horário. Começando em 5^o e terminando em 85^o, são 17 rotações. Então são $17 \times 10 = 170$ equações e 64 variáveis.

Você vai receber 2 arquivos: o primeiro de nome `Fk37COEF.myd` que está publicado em `algoritmovivo.com`, pasta "atividades"e que contém os coeficientes do sistema linear que deverá usar para reconstituir a imagem. Estes coeficientes formam uma matriz de 170 linhas e 64 colunas, contendo números reais entre 0 e 1. Esta matriz é a mesma para todos pois ela está associada a este exercício. Trata-se da matriz A em $Ax = B$.

Além deste arquivo você deve recuperar o arquivo

`Fk31ES05.myd`

que contém a coluna de termos independente do seu exercício (o B em $Ax = B$) que é único para você.

Depois de resolver o sistema linear (usando o método dos mínimos quadrados), responda o que tem nos pixels das linhas 1..4 (primeira é a superior) na coluna 8 (a última).

Para testar seus programas, confira os arquivos `Fk37EXEn.myd` no mesmo local: o primeiro tem apenas as 2 primeiras linhas iguais a 1, as demais são zero. O segundo é a negação do primeiro. O terceiro é aleatório, e as posições pedidas (últimas colunas, primeira à quarta linha) são: 1 1 0 1. Finalmente o exemplo 4 traz apenas as colunas 3, 4 e 5 valendo 1. As demais valem zero.

Leitura de dados em Python

Para ter acesso aos dados, você vai precisar baixar os arquivos publicados em seu computador e processar um programa Python semelhante a este:

```
import numpy as np
def leitcoef():
    coef=np.zeros((680,16),float)
    ref=open("f:/n/k37/fk37coef.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        lin=lin.split()
        for j in range(0,16):
            coef[i,j]=float(lin[j])
        i=i+1
        lin=ref.readline()
    coef.resize([170,64])
    return(coef)
a=leitcoef()
print(a)
```

Ao final deste trecho a será uma matriz contendo os coeficientes A da matriz de solução deste problema.

A seguir, voce vai precisar ler seus coeficientes particulares. Faça

```
def leiti():
    ti=np.zeros((170),float)
    ref=open("x:/xxx/fk3700nn.myd","r")
    lin=ref.readline()
    i=0
    while (0!=len(lin)):
        ti[i]=float(lin)
        i=i+1
        lin=ref.readline()
    return(ti)
b=leiti()
print(b)
```

Finalmente, a solução do sistema linear usa a funçao `lstsq` do pacote linalg de numpy. Acompanhe

```
c=np.linalg.lstsq(a,b,rcond=None)
ima=np.zeros((8,8),float)
k=0
xx=c[0]
for i in range(0,8):
    for j in range(0,8):
        ima[i,j]=abs(float("%8.3f" % xx[k]))
        k=k+1

print(ima)
```

Responda aqui:

L1 C8	L2 C8	L3 C8	L4 C8

Para saber mais Consulte:

DEWDNEY, A.K. **The Turing Omnibus**. New York, Freeman Co. 1989.
Kepler de Souza Oliveira Filho. In http://www.if.ufrgs.br/ast/ med/imagens/ imagens.htm
GUSTAVSON, Katarina. **The Role of Linear Algebra in Computed Tomography** In https://raysforexcellence.squarespace.com/.