Day 3

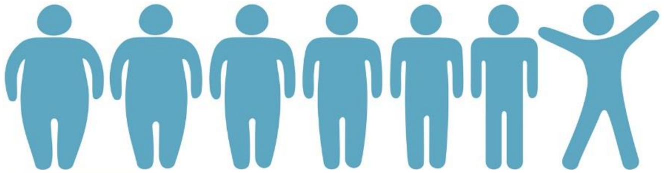
スパース解法

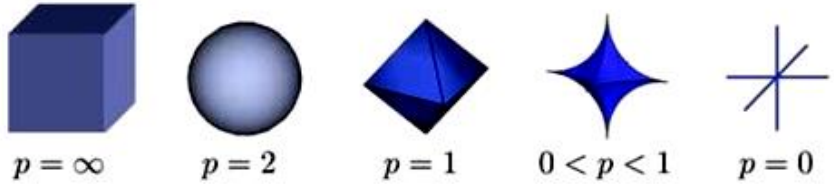
Sparse solvers

Two major approaches to sparse solution

- Combinatorial greedy algorithm
  iteratively selects nonzero candidates that reduce the current residual.
  - MP (Matching Pursuit) / OMP (Orthogonal MP) / ROMP (Regularized OMP) / StOMP (stagewise OMP) / gOMP (Generalized OMP) / Subspace pursuit for CS

- Convex relaxation
  replaces the ℓ_0-norm with the ℓ_1-norm and applies mathematical optimization techniques.
  - ISTA (Iterative soft thresholding algorithm) / FISTA (Fast ISTA)
  - ADMM (alternating direction method of multipliers) / PDS (primal dual splitting)





貪欲法

問　なるべく少ない枚数のコインで623円を作る手順を説明せよ.
ただし，説明相手は小学校1年生で，
数の大小と，足し算，引き算しか知らないものとする.

OMP

Orthogonal matching pursuit

$$\underset{\boldsymbol{x}}{\text{Minimize}}\ \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Repeat:

- pick up $\boldsymbol{a}^{(s)}$ which is most similar to the residual $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$, and update $T \leftarrow T \cup \{s\}$;
- estimate the nonzeros $\boldsymbol{x}_T$ by least squares ($\boldsymbol{r} \perp \boldsymbol{A}\boldsymbol{x}$);

```
def OMP(A, b, tol=1e-5, maxnnz=np.inf):
    m, n = A.shape
    supp = []
    x = np.zeros(n)
    r = b.copy()
    while len(supp) < maxnnz and linalg.norm(r) > tol:
        s = np.argmax(np.abs( A.T.dot(r) ))
        supp.append(s)
        Asupp = A[:,supp]
        x[supp] = np.linalg.lstsq(Asupp, b)[0]
        r = b - Asupp.dot(x[supp])
    return x
```

例

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ -2 \\ -1 \\ -1 \end{bmatrix}$$

```
x = [0.00, 0.00, 0.00, 0.00, 0.00]'
r = [2.00, -2.00, -1.00, -1.00]'
c = [2.00, -3.00, -4.00, -1.00, -1.00]'
s = 3
T = [3]
x = [0.00, 0.00, -0.57, 0.00, 0.00]'
r = [2.57, -0.86, -0.43, -0.43]'
c = [2.57, -1.29, -0.00, -0.43, -0.43]'
s = 1
T = [3  1]
x = [3.00, 0.00, -1.00, 0.00, 0.00]'
r = [-0.00, 0.00, 0.00, 0.00]'
```

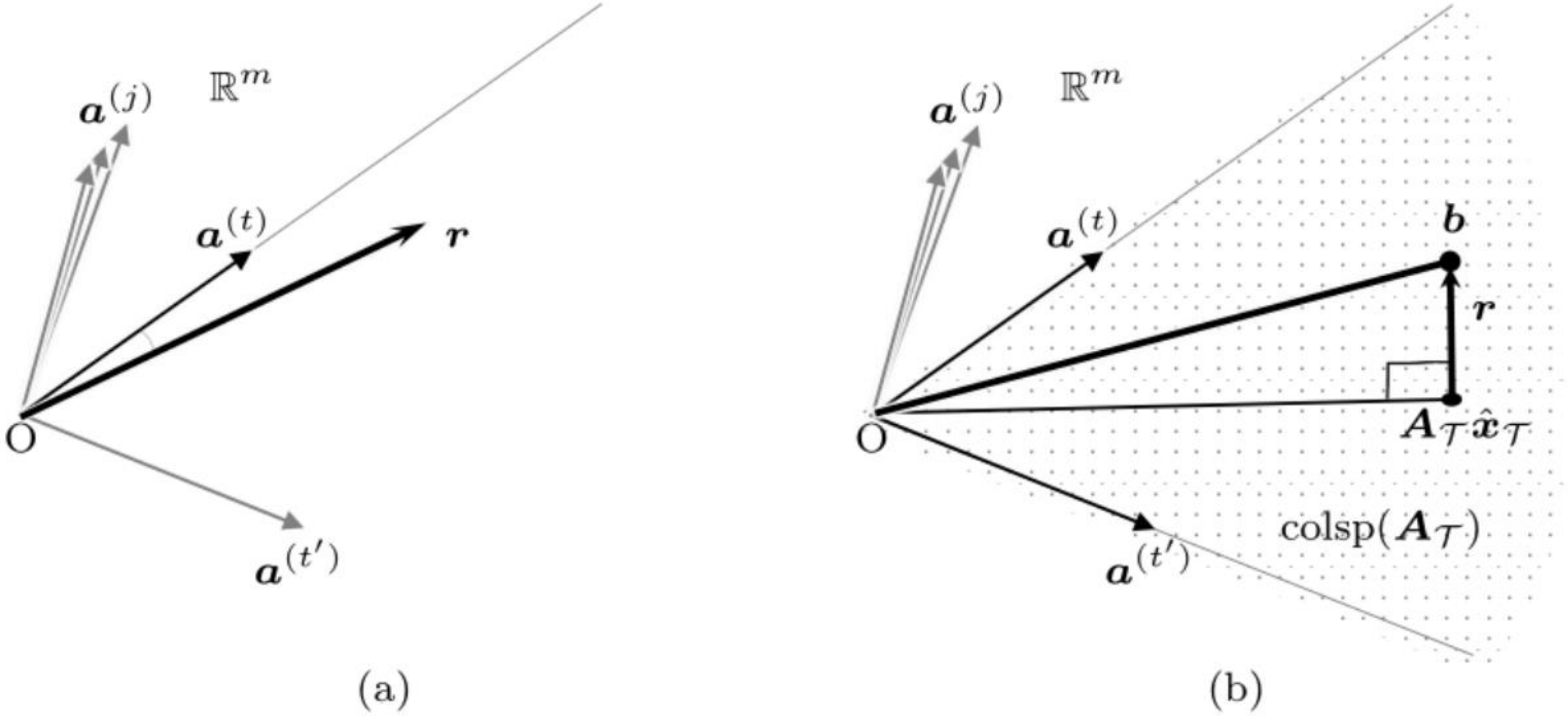


図 3.6 OMP における原料の選出と残差の更新．(a) MP と同様に，残差 $\boldsymbol{r}$ に方向が最も類似する原料 $\boldsymbol{a}^{(t)}$ を選出し，t を台 $\mathcal{T}$ に追加する．(b) $\boldsymbol{A}_{\mathcal{T}}\hat{\boldsymbol{x}}_{\mathcal{T}} \in \mathrm{colsp}(\boldsymbol{A}_{\mathcal{T}})$ は，選出済みの原料によるデータ $\boldsymbol{b}$ の近似である．最小の ℓ_2 誤差を達成する近似は，$\boldsymbol{b}$ の $\mathrm{colsp}(\boldsymbol{A}_{\mathcal{T}})$ への正射影である．更新後の残差 $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}_{\mathcal{T}}\hat{\boldsymbol{x}}_{\mathcal{T}}$ は $\mathrm{colsp}(\boldsymbol{A}_{\mathcal{T}})$ と直交するので，原料の選出が重複することはない．

台

【定義】（ベクトルの台）

非ゼロ成分の番号（添え字）の集合をベクトルの**台**（support）と呼ぶ.

例： $\boldsymbol{x} = [0,2,0,0,1]^\top$ の台は $\mathcal{S} =$

【定義】（列や成分の抜き出し）

- 番号の集合 $\mathcal{T}$ で指定して $\boldsymbol{A}$ の列を抜粋した行列を $\boldsymbol{A}_\mathcal{T}$ と記す.
  $\boldsymbol{A}_\mathcal{T}$ のサイズは m 行 $|\mathcal{T}|$ 列である（ $|\mathcal{T}|$ は集合 $\mathcal{T}$ の要素数）.
- 同様に, $\mathcal{T}$ で指定して $\boldsymbol{x}$ の成分を抜粋したベクトルを $\boldsymbol{x}_\mathcal{T}$ と記す.

例： $\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix}$, $\boldsymbol{x} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$, $\mathcal{T} = \{1, 3\}$ ならば， $\boldsymbol{A}_\mathcal{T} =$ $\boldsymbol{x}_\mathcal{T} =$

https://colab.research.google.com/github/tsakailab/spmlib
/blob/master/demo/eg01_GreedyAlgorithms.ipynb

今日は
ここまで