Kaczmarz’ Method (review)

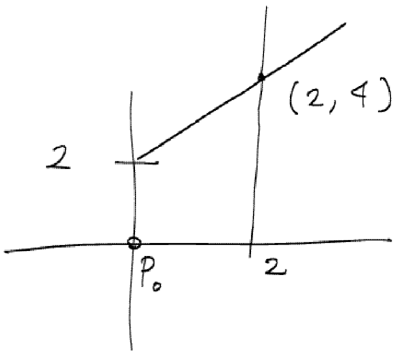
$p = 0$

for each iteration i

for each row a_j^T of A

$$p \leftarrow p + \frac{(b_j - a_j^T p)\, a_j}{a_j^T a_j}$$

Example



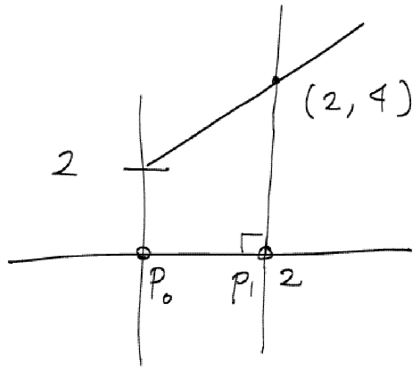
$$Ax = b$$

$$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$p_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$a_1^T = [1 \quad 0] \qquad b_1 = [2]$

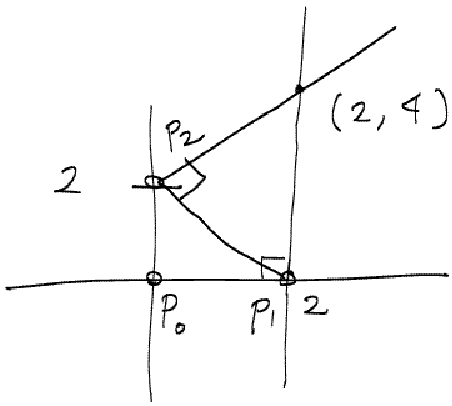
$$\frac{(b_1 - a_1^T p)\, a_1}{a_1^T a_1} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$



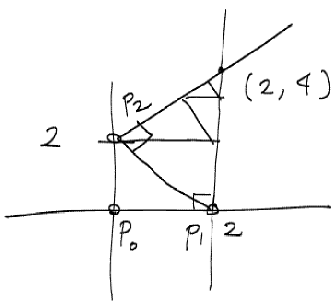
$$p = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad a_2^T = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 2 \end{bmatrix}$$

$$p + \frac{b_2 - a_2^T p}{a_2^T a_2} a_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} + \frac{2+2}{2} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$



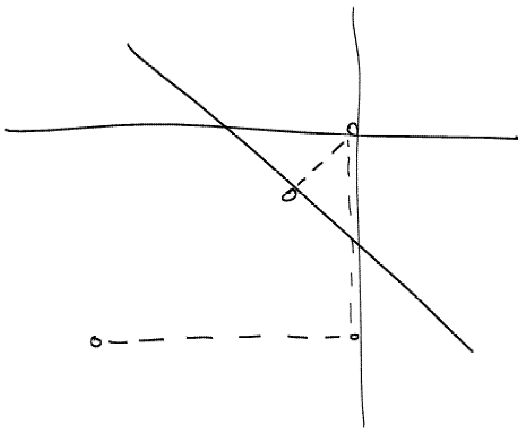
And so on…



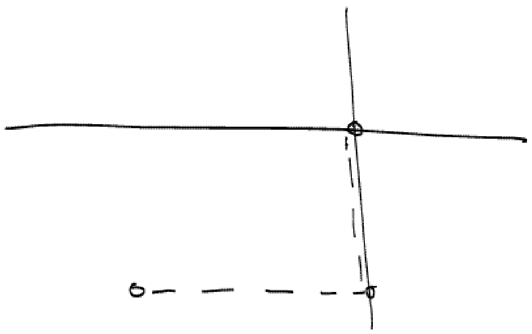
Convergence of Kaczmarz

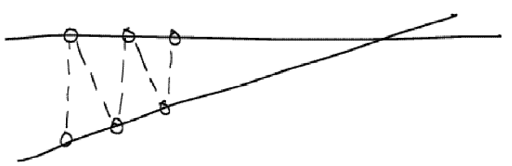
If there is a unique solution, then Kazmarz alg converges to the solution. [Tanabe 71].

However, generally there is not a unique solution.



Angles between affine spaces affect convergence.



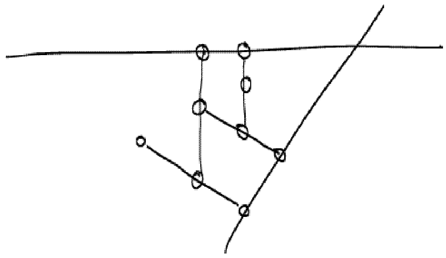


Methods for improving convergence:

- orthogonalization
- careful choice of projection order

Relaxation

It is possible to improve the image quality using relaxation.



$$p \leftarrow p + \alpha \frac{b_i - a_i^T p}{a_i^T a_i} a_i$$

SIRT (1972)

“Simultaneous Iterative Reconstruction Technique”
Not quite Cimmino’s method, but similar…
Gather together changes to a voxel from all rays, then update by the average

Initialize $p^{(0)}$
do until convergence
  $q = 0, \; n = 0$
  for each row of A
    $q \leftarrow q + \frac{b_i - a_i^T p}{a_i^T a_i} a_i$
    $n \leftarrow n + \tau(a)$
  $p \leftarrow p + q/n$

$\tau(a)$ selects non-zero elements of a.

Cimmino’s Method (1938)

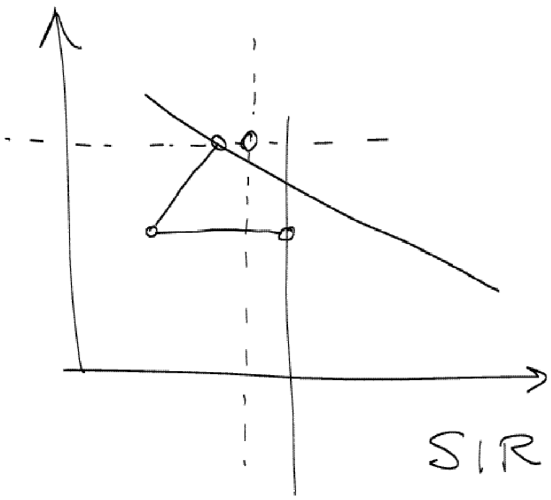
initialize P

do until convergence

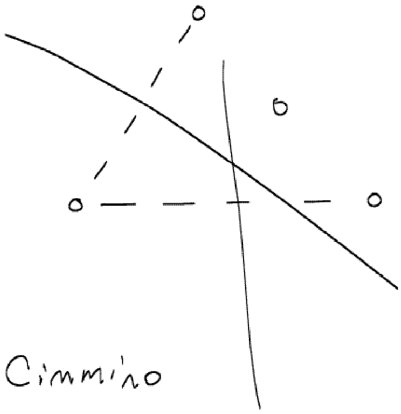
for each row i of A

$$q_i = p + \alpha \frac{b - a_i^T p}{a_i^T a_i} a_i$$

$$p \leftarrow \frac{1}{N} \sum_i q_i$$



SIRT

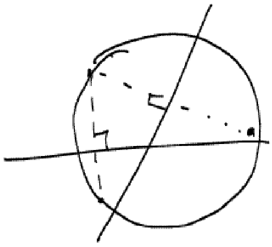


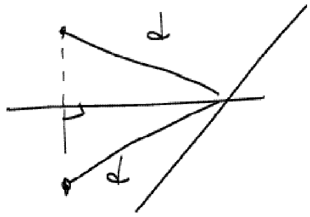
Cimmino

Convergence of Cimmino’s method

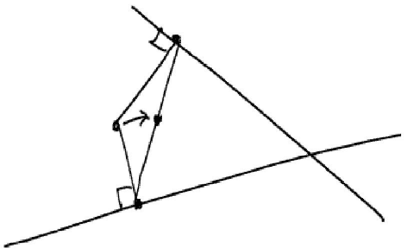
Cimmino’s method was known to converge in 1938 (for consistent data)

Proof is based on the knowledge that the solution lies at the center of an n-sphere and the initial position and the projections lie on the surface





Occasionally Cimmino’s algorithm is defined w/o factor of 2



SART (1984)

Simultaneous Algebraic Reconstruction Technique

for each projection

  for each row i in projection

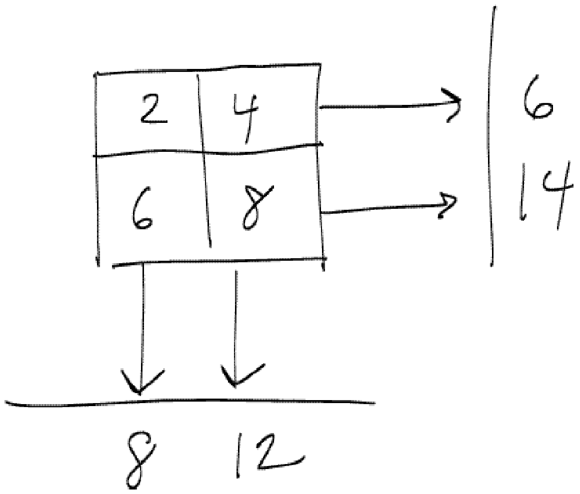
$$q \leftarrow q + \frac{b_i - a_i^T p}{a_i^T a_i} a_i$$

  Normalize correction

$$q_i \leftarrow q_i \Big/ \sum_j A_{ij}$$

  $p \leftarrow p + q$

Example



$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} 6 \\ 14 \\ 8 \\ 12 \end{bmatrix}$$

Projection #1

$$g \leftarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{(6-0)}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 0 \\ 0 \end{bmatrix}$$

$$g \leftarrow \begin{bmatrix} 3 \\ 3 \\ 0 \\ 0 \end{bmatrix} + \left(\frac{14-0}{2}\right) \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 7 \\ 7 \end{bmatrix}$$

$$\sum_j A_{ij} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$g \Big/ \sum_j A_{ij} = \begin{bmatrix} 3 \\ 3 \\ 7 \\ 7 \end{bmatrix}$$

Projection #2

$$a_3^T p = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \\ 7 \\ 7 \end{bmatrix} = 10$$

$$q \leftarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \left(\frac{8-10}{2}\right) \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$a_4^T p = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \\ 7 \\ 7 \end{bmatrix} = 10$$

$$q \leftarrow \begin{bmatrix} -1 \\ 0 \\ -1 \\ 0 \end{bmatrix} + \left(\frac{12-10}{2}\right) \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

Matrix format

for each row i

$$q \leftarrow q + \frac{b_i - a_i^T p}{a_i^T a_i} a_i$$

$$q = \frac{b_1 - a_1^T p}{a_1^T a_1} a_1 + \frac{b_2 - a_2^T p}{a_2^T a_2} a_2 + \cdots$$

$$\underbrace{\begin{bmatrix} \frac{1}{a_1^T a_1} & & \\ & \ddots & \\ & & \frac{1}{a_N^T a_N} \end{bmatrix}}_{\text{denominator}} \underbrace{\left(\begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} - \begin{bmatrix} a_1^T \\ \hline \\ \hline a_N^T \end{bmatrix} \begin{bmatrix} p \end{bmatrix} \right)}_{\text{numerator}}$$

$$\begin{bmatrix} a_1 & \cdots & a_N \end{bmatrix} \begin{bmatrix} \frac{1}{a_1^T a_1} & & \\ & \ddots & \\ & & \frac{1}{a_N^T a_N} \end{bmatrix} \left(\begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} - \begin{bmatrix} a_1^T \\ \hline \\ \hline a_N^T \end{bmatrix} \begin{bmatrix} p \end{bmatrix} \right)$$

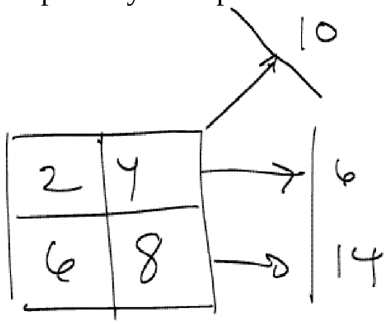
which yields:

$$p \leftarrow p + A^T D\,(b - Ap)$$

CAV (2001)

"Component Averaging"
Uses oblique (not orthogonal) projections

Easiest to explain by example



$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 6 \\ 14 \\ 10 \end{bmatrix}$$

Define s_j as number of non-zero elements in $j^{\underline{th}}$ column

$$s = [1 \;\; 2 \;\; 2 \;\; 1]$$

$$\text{Define } g_{ij} = \begin{cases} 1/\sqrt{s_j} & \text{if } A_{ij} \neq 0 \\ 0 & \text{ow} \end{cases}$$

$$G = \begin{bmatrix} 1 & 1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \end{bmatrix}$$

for each row i of A

$$q \leftarrow q + \frac{b - a_i^T p}{(Ga_i)^T (Ga_i)} a_i$$

$$q = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{6 - 0}{3/2} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 0 \\ 0 \end{bmatrix}$$

DROP (2005)

"Diagonally relaxed orthogonal projections"

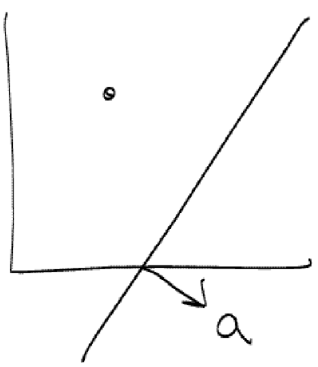
Let $S = \mathrm{diag}(s)$

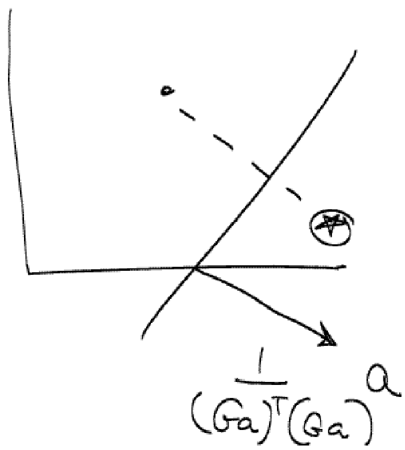
For each row i of A

$$q \leftarrow q + \frac{(b_i - a_i^T)}{a_i^T a_i} S a_i$$

Fundamentally different from CAV

CAV





DROP