Data normalization

What if we want to apply 2+ transformations to data? e.g. a translation and a scaling?

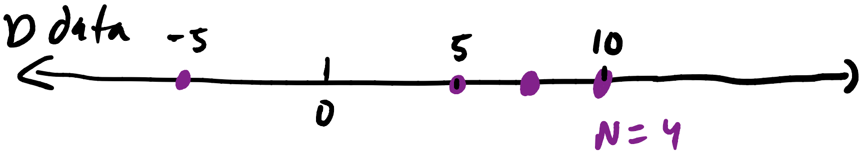
Normalization is one operation that requires **both** T and S.

Normalization for one variable:

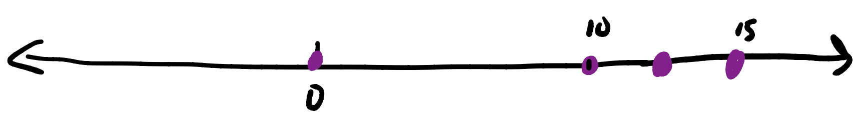
$\boxed{\text{Basic idea}}$ Scale data so all values in range [0, 1]

↑ ↗ inclusive

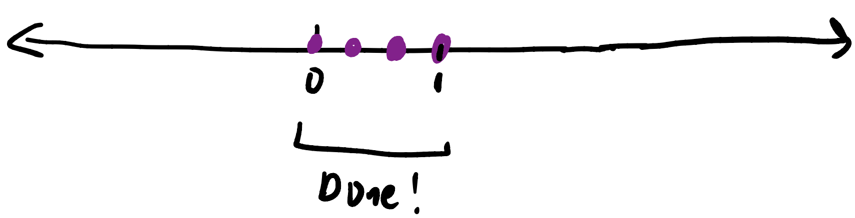
1D data



1) subtract min from all samples:



2) Divide by range = max - min.



$\boxed{\text{Formula}}$ $$\frac{\overset{\text{data}}{\vec{X}} - \text{min}}{\underbrace{\text{max} - \text{min}}_{\text{range}}} = \vec{X}_{norm}$$

Where is the translation? $\boxed{x - \min}$

ie. translate data by $-\min$.

Where is the scaling? $\boxed{\frac{1}{\max - \min} = \frac{1}{\text{range}}}$

ie. divide data by range [after translating]

Now thinking of >1 variables, how would we setup the matrix multiplication to apply both T & S to D?

$$A' = (T @ S @ A.T).T \quad \text{or} \quad A' = (S @ \underbrace{T @ A.T}).T$$

translation gets applied to A first, then we do S @(result)

How do we fill in T & S to do normalization?

Option 1 : Global normalization (normalize together)

For T : translate by overall min over all data
[without homogenous coord]

For S : Scale by range over all data
[without homogenous coord]

Example : Temperature data

$$A = \begin{array}{cc} \text{ME} & \text{FL} & \\ \end{array}$$

$$A = \begin{bmatrix} 32 & 100 & 1 \\ 5 & 90 & 1 \\ 10 & 80 & 1 \end{bmatrix}$$

2D data $\Rightarrow$ $T = \begin{bmatrix} 1 & 0 & \boxed{t_x} \\ 0 & 1 & \boxed{t_y} \\ 0 & 0 & 1 \end{bmatrix}$ need these!

Translate all variables by $\underline{-\text{global min}} = 5$

$$T = \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T@A.T = \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 32 & 5 & 10 \\ 100 & 90 & 80 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 27 & 0 & 5 \\ 95 & 85 & 75 \\ 1 & 1 & 1 \end{bmatrix}$$

ME + FL temps both shifted by -5

Now for the Scaling:

need these!

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

we scale by $\dfrac{1}{\text{global range}}$: $\dfrac{1}{\underbrace{\text{global max}}_{100} - \underbrace{\text{global min}}_{5}}$

$$= \frac{1}{95}$$

$$S@T@A.T = \begin{bmatrix} 1/95 & 0 & 0 \\ 0 & 1/95 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 27 & 0 & 5 \\ 95 & 85 & 75 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 27/95 & 0 & 5/95 \\ 1 & 85/95 & 75/95 \\ 1 & 1 & 1 \end{bmatrix}$$

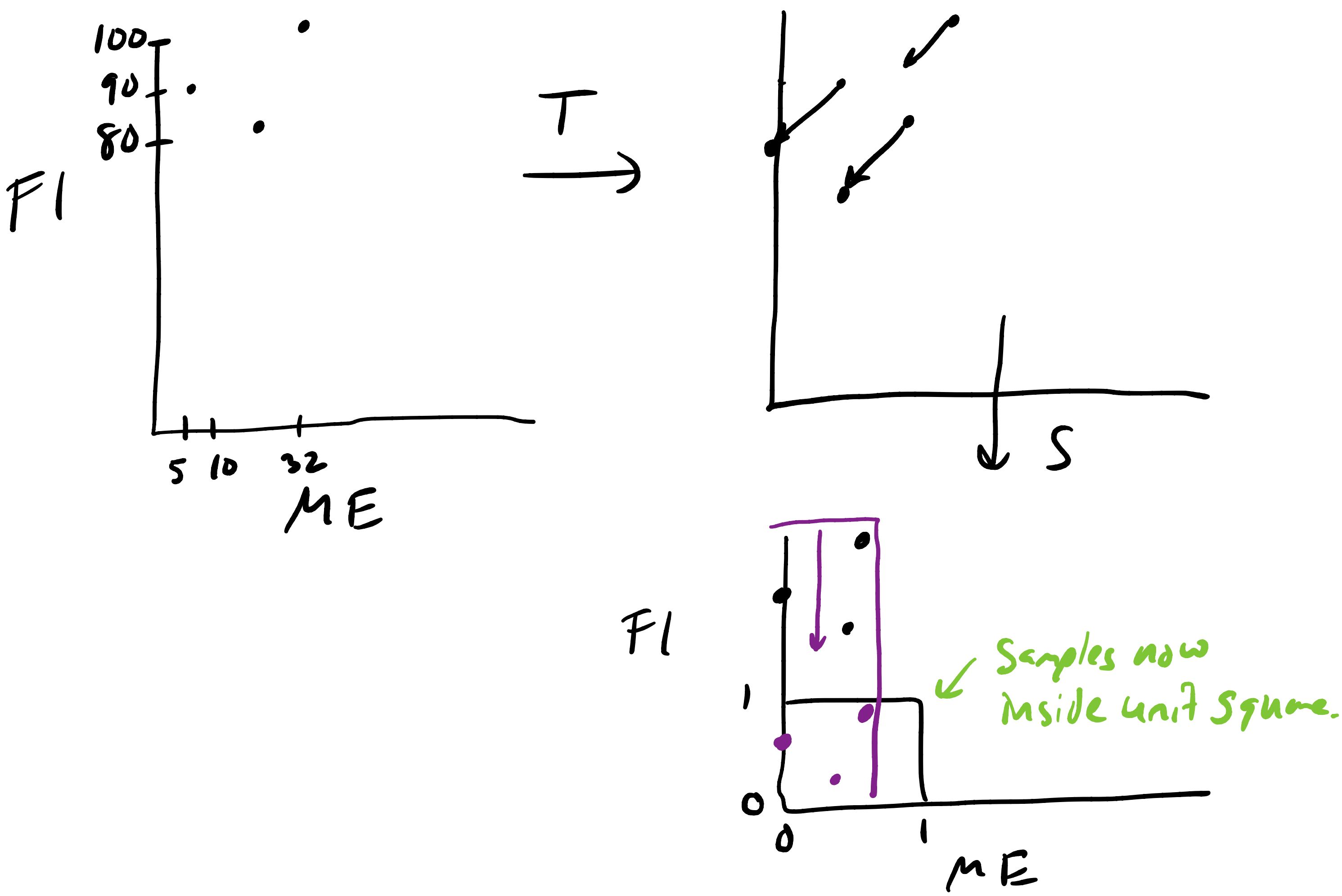
$$\boxed{(S@T@A.T).T = \begin{bmatrix} 27/95 & 1 & 1 \\ 0 & 85/95 & 1 \\ 5/95 & 75/95 & 1 \end{bmatrix}}$$

☆ all data now between 0 + 1.

Sanity check

Across whole matrix, there should be 1 at overall max + 0 at the overall min — other values should be in between.

Geometry of global normalization:



Option 2 : Per-Variable normalization (normalize separately)

$\boxed{\text{For } T}$: translate by each var's min
[without homogenous coord]

$\boxed{\text{For } S}$: Scale by range each variable.
[without homogenous coord]

$$A = \begin{array}{c} \begin{array}{ccc} ME & FL & \end{array} \\ \begin{bmatrix} 32 & 100 & 1 \\ 5 & 90 & 1 \\ 10 & 80 & 1 \end{bmatrix} \end{array}$$

need these!

$$T = \begin{bmatrix} 1 & 0 & \boxed{t_x} \\ 0 & 1 & \boxed{t_y} \\ 0 & 0 & 1 \end{bmatrix}$$

$\Rightarrow$ Translate by $-$min of $\underbrace{ME}_{5}$ and $\underbrace{FL}_{80}$ separately:

$$T@A.T = \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -80 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 32 & 5 & 10 \\ 100 & 90 & 80 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 27 & 0 & 5 \\ 20 & 10 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

need these

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\Rightarrow$ Scale by $\frac{1}{\text{range}}$ for ME & FL computed separately

$\Rightarrow$ ME max − min = 32 − 5 = 27

FL max − min = 100 − 80 = 20

$$S@T@A.T = \begin{bmatrix} 1/27 & 0 & 0 \\ 0 & 1/20 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 27 & 0 & 5 \\ 20 & 10 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 5/27 \\ 1 & 10/20 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

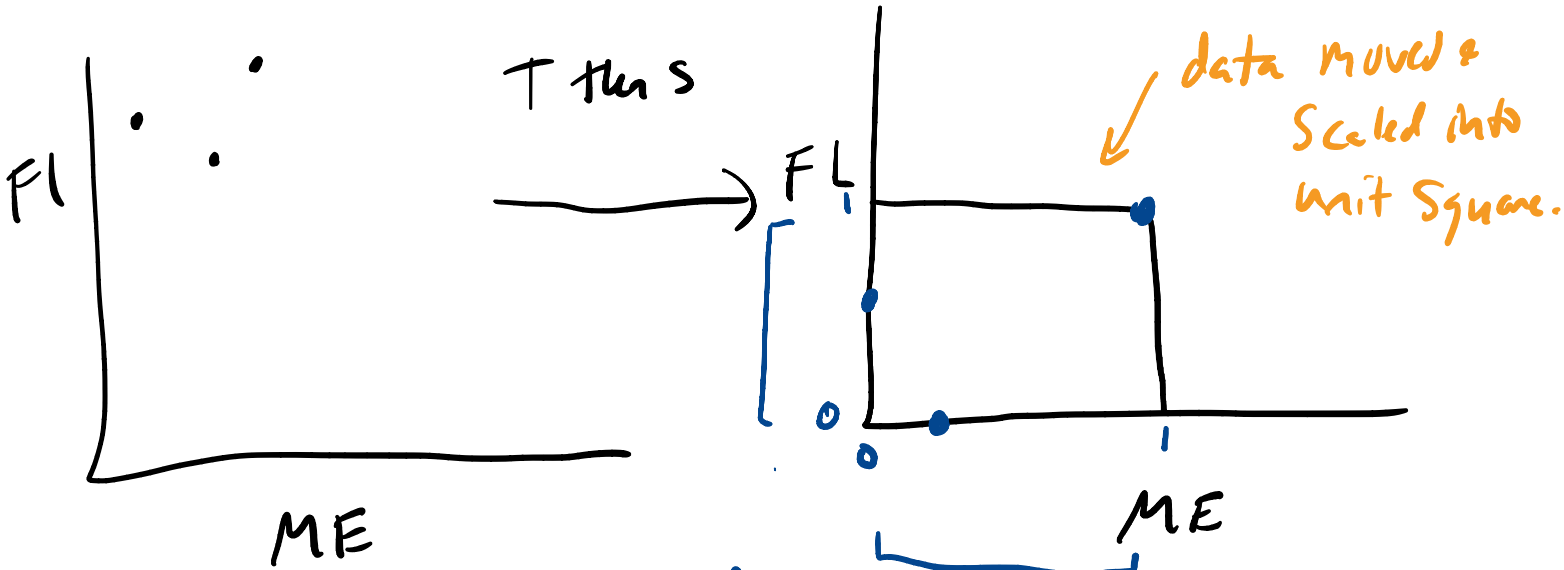
$$\boxed{(S@T@A.T).T = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 10/20 & 1 \\ 5/27 & 0 & 1 \end{bmatrix}}$$

☆ all data now between 0 & 1.

Sanity check A one and zero in each column — one set per variable

[unless there are ties for min/max]

Geometry of normalize seperately:



★ data spans [0,1] range in **each** variable after normalization

performing 2+ transformations (e.g. T then S) called **Compound transformation**.

Because both T & S have the same shapes, $(M+1, M+1)$ [e.g. (3,3) for 2D data]

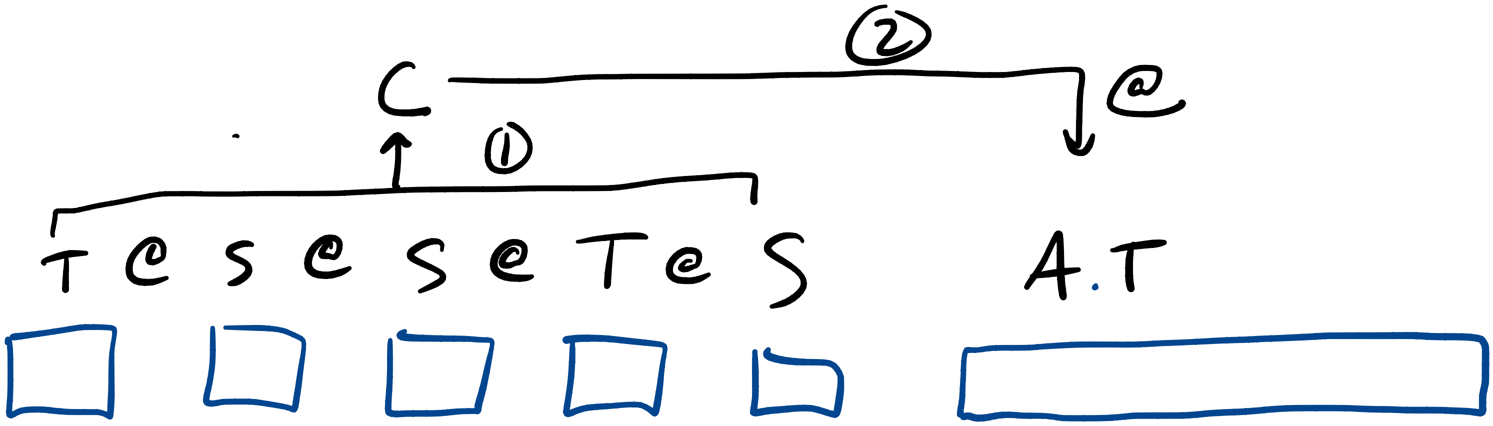
we can multiply them in advance and get **one** $(M+1, M+1)$ matrix C.

e.g. normalization: $C = S @ T$

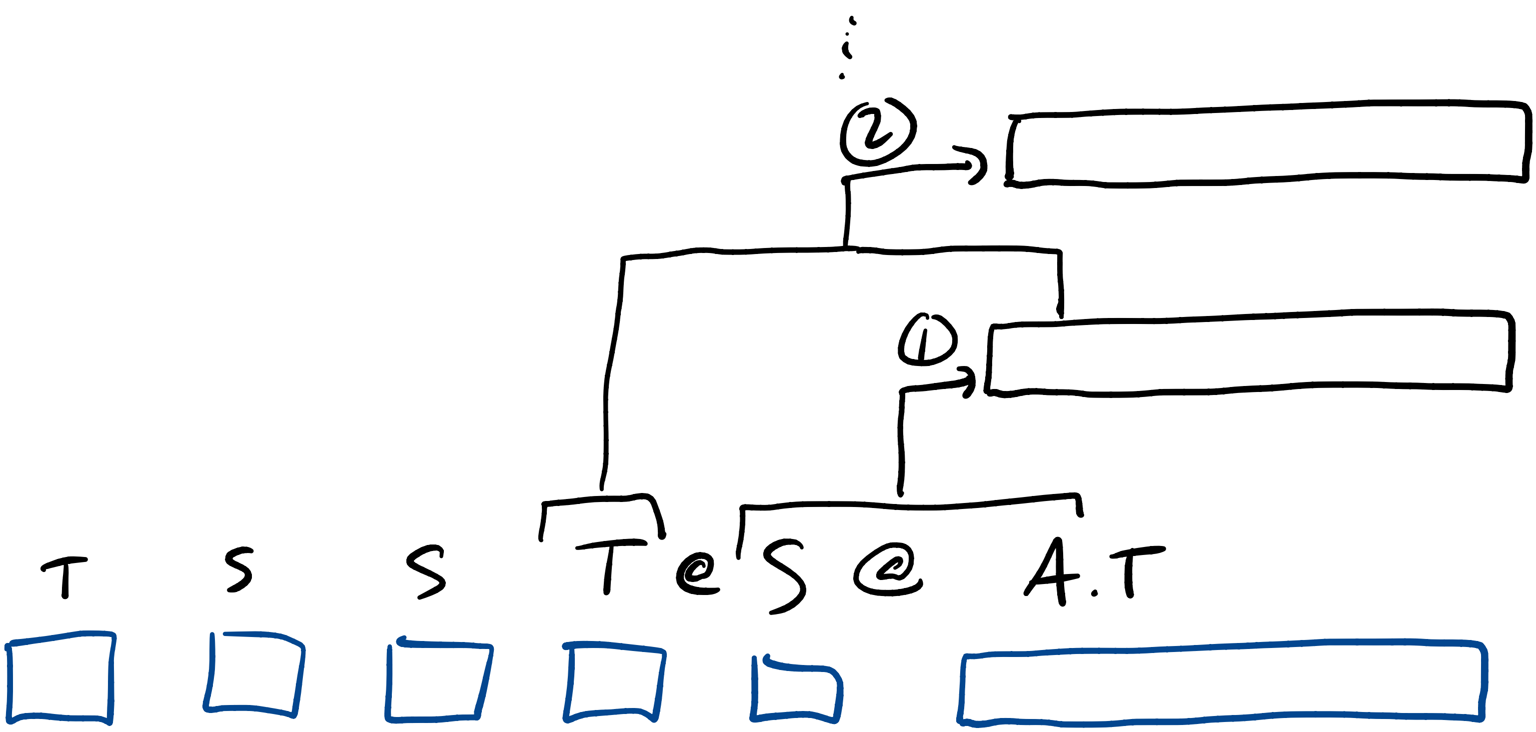
then we always do: $A' = (C @ A.T).T$

Which would be more efficient:

1) Multiplying all the transformation matrices first to get C ⇒ then do one transformation on A?



2) Multiply each transformation matrix with A.T one-by-one?



$\boxed{\#1}$! apply **all** transformations with **one** matrix multiplication with data matrix

Why is computing C desirable? If $\overset{\text{e.g. } 1000}{N} > \overset{4}{M}$ (many cases)

- Computing lots of 4x4 matrix multiplications to get C is **fast**. [#1]
- Doing lots of 1000 x 4 matrix multiplications **slow**. [#2]

☆ When computing C, need to arrange transformation matrices in proper order:

$$C = S @ T \neq T @ S$$