DMIP - Exercise:

RANSAC

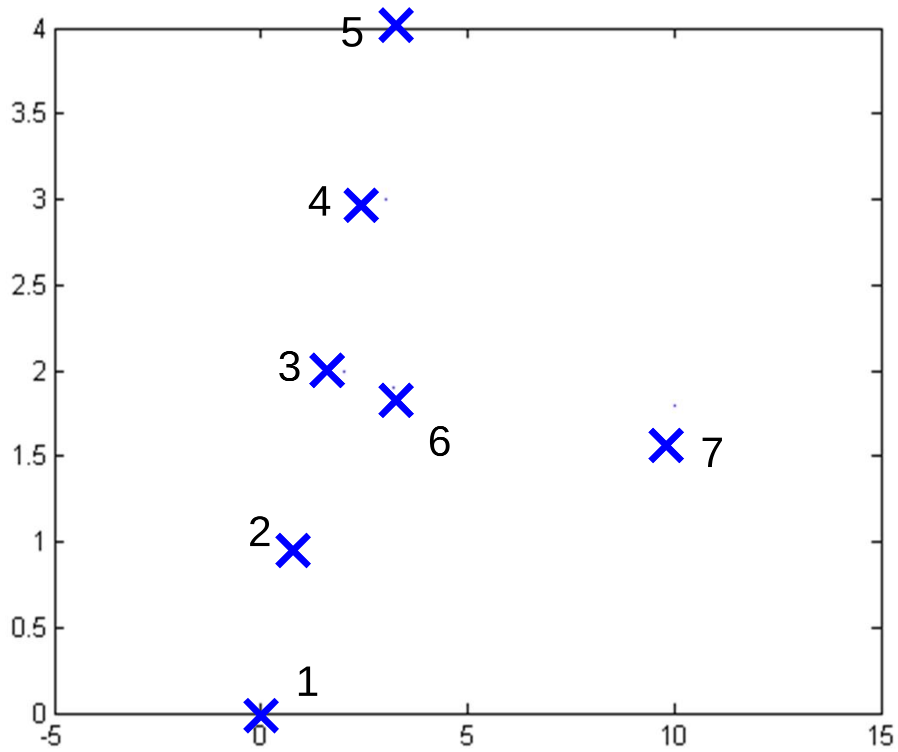
Yan Xia, Marco Bögel
Pattern Recognition Lab (CS 5)

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Problem in calibration: inaccuracies in observations and outliers.

- Badly localized points (noise)
- Wrong correspondence

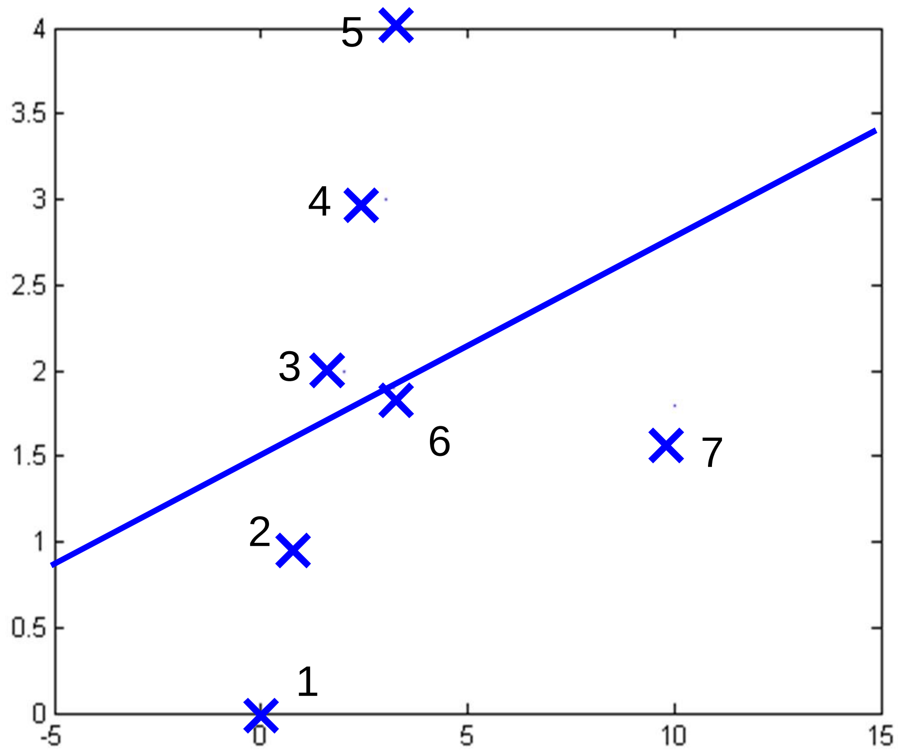
Linear Regression



Problem in calibration: inaccuracies in observations and outliers.

- Badly localized points (noise)
- Wrong correspondence

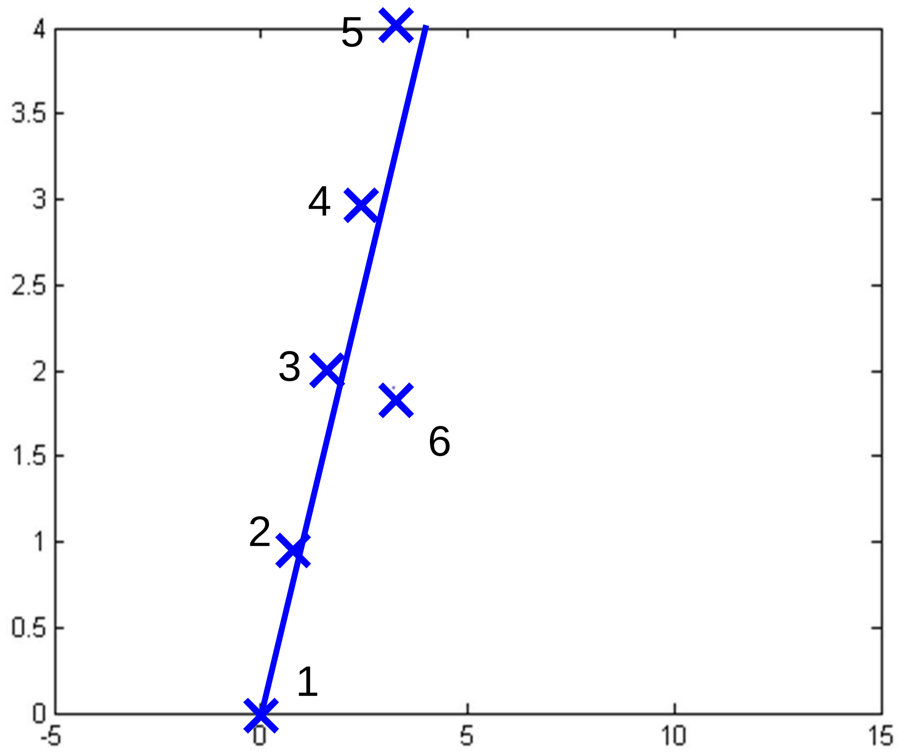
Linear Regression



Problem in calibration: inaccuracies in observations and outliers.

- Badly localized points (noise)
- Wrong correspondence

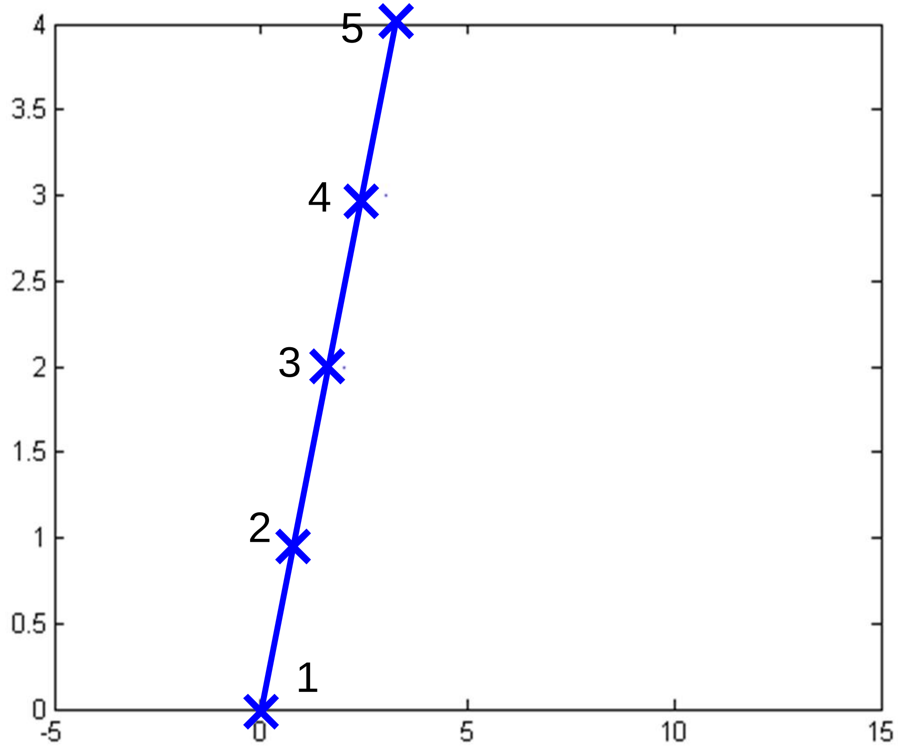
Linear Regression



Problem in calibration: inaccuracies in observations and outliers.

- Badly localized points (noise)
- Wrong correspondence

Linear Regression



RANSAC – RANdom Sample Consensus

RANSAC assumes that a model built with a minimum number of data points for this model **does not contain outliers**.

Algorithm:

- Determine the minimum number n_{mdl} of data points required to build the model
  → A line is completely defined by two points → $n_{mdl} = 2$

- For n_{it} iterations do
  - a) Choose randomly n_{mdl} points out of your data to estimate the model
  - b) Determine the error of the current model using all data points

- Choose model with lowest error

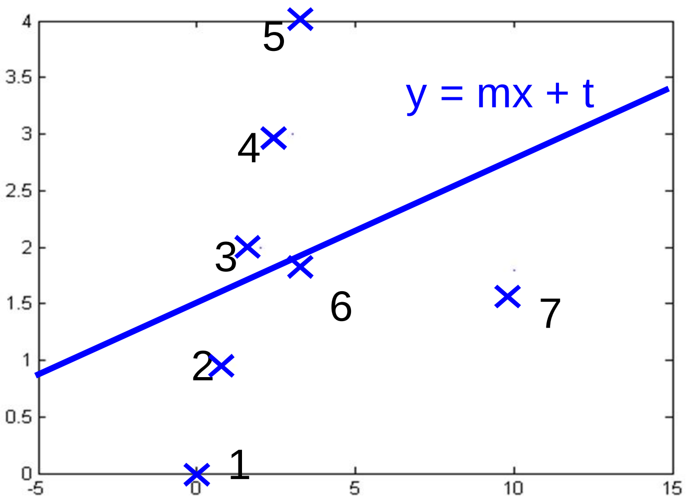
RANSAC

Task: complete the function `fitline:` This will be used for fitting a line through a set of points.

Find the line parameter *m* and *t*, so that all points (x_i, y_i), i = 1,…,7, approximately fulfill the line equation $y_i = mx_i + t$

→ Solve the following optimization problem

$$\left\| [X\ 1] \cdot \begin{pmatrix} m \\ t \end{pmatrix} - Y \right\| = \left\| M \cdot \begin{pmatrix} m \\ t \end{pmatrix} - Y \right\| \to 0$$

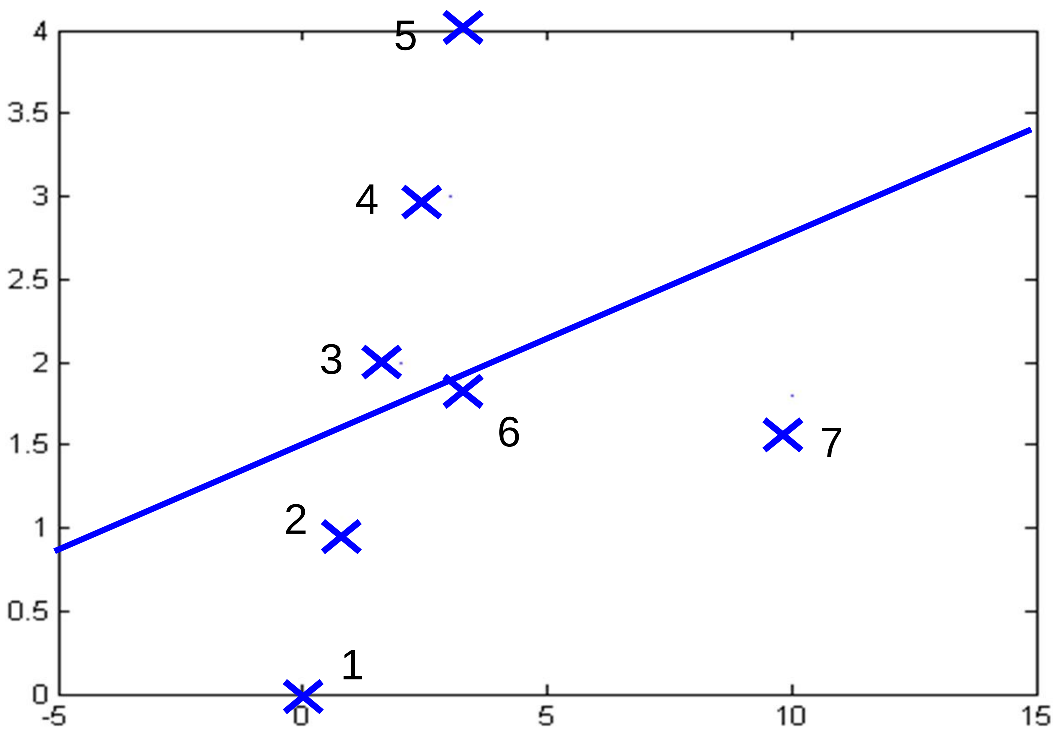


The least square solution of this equation is given (**Moore-Penrose pseudo-inverse)**

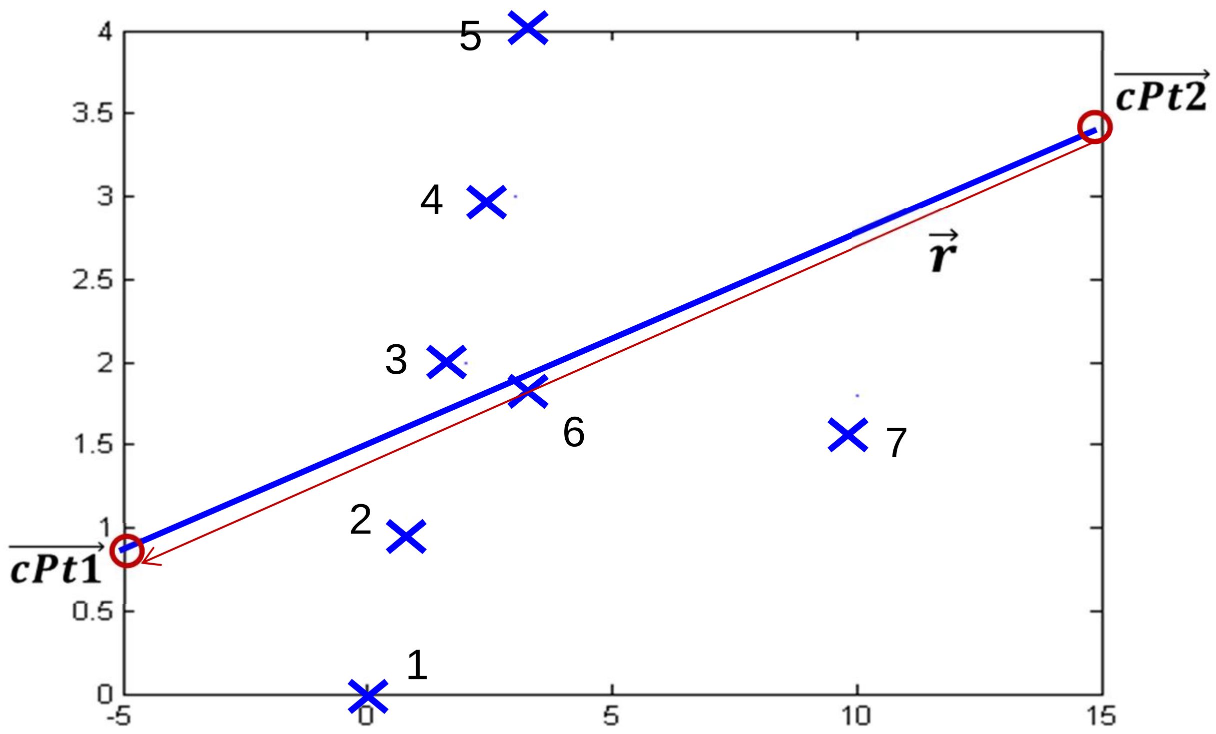
$$\begin{pmatrix} m \\ t \end{pmatrix} = M^{\dagger} Y$$

RANSAC

Task: `lineerror`: This will be our specialized `errFct` for our line model `mdl` considering all samples in `pts`. Think about a proper error metric.

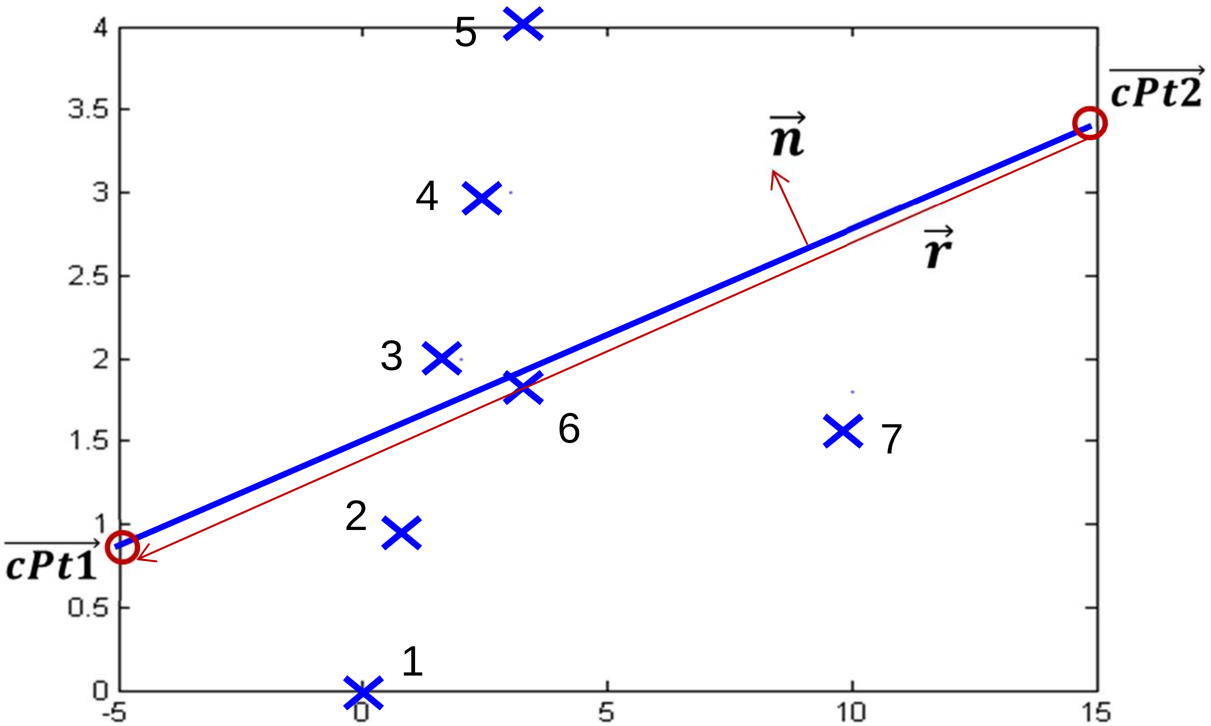


4
3.5
3
2.5
2
1.5
1
0.5
0
-5
0
5
10
15
1
2
3
4
5
6
7



1) Pick two points on the line and calculate direction:

$$\vec{r} = c\vec{P}t2 - c\vec{P}t1$$



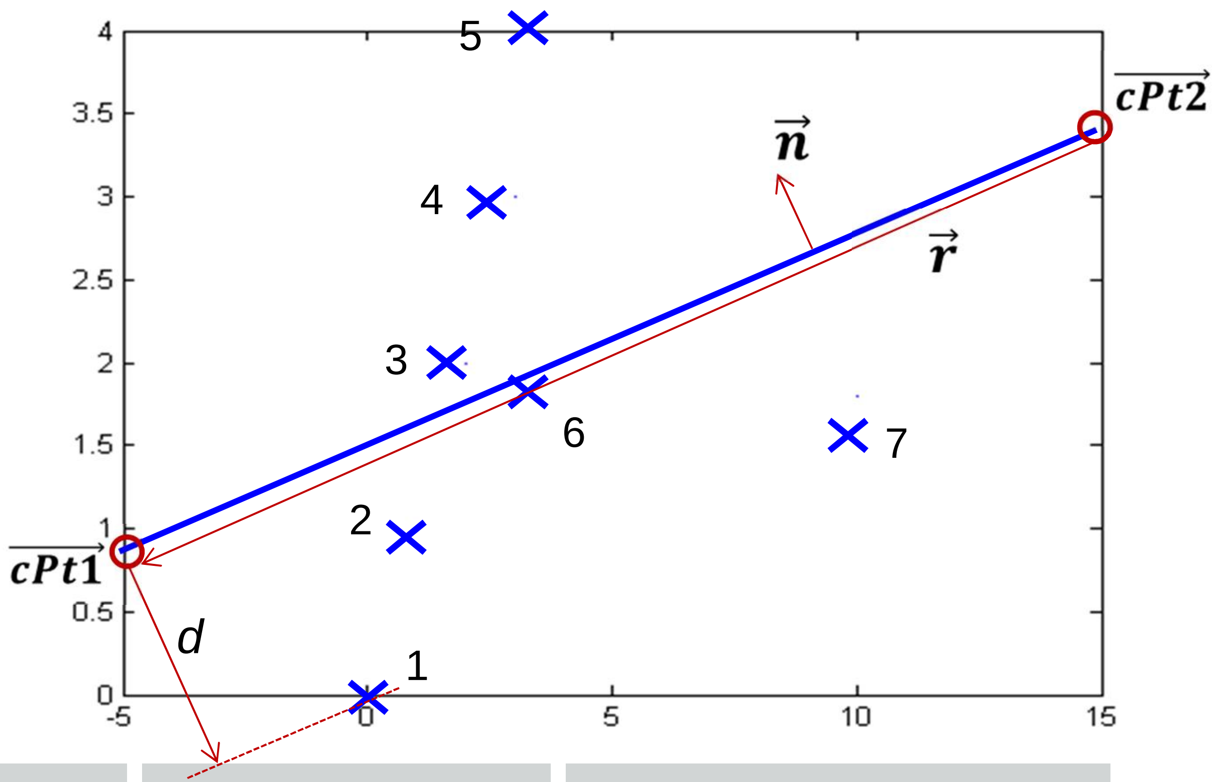
1) Pick two points on the line and calculate direction:

$$\vec{r} = c\vec{P}t2 - c\vec{P}t1$$

2) Calculate normal vector to direction and normalize it:

$$\vec{n} = \begin{pmatrix} -y_{\vec{r}} \\ x_{\vec{r}} \end{pmatrix}$$

$$\vec{n} = \vec{n}/\text{norm}(\vec{n})$$



1) Pick two points on the line and calculate direction:

$$\vec{r} = c\vec{P}t2 - c\vec{P}t1$$

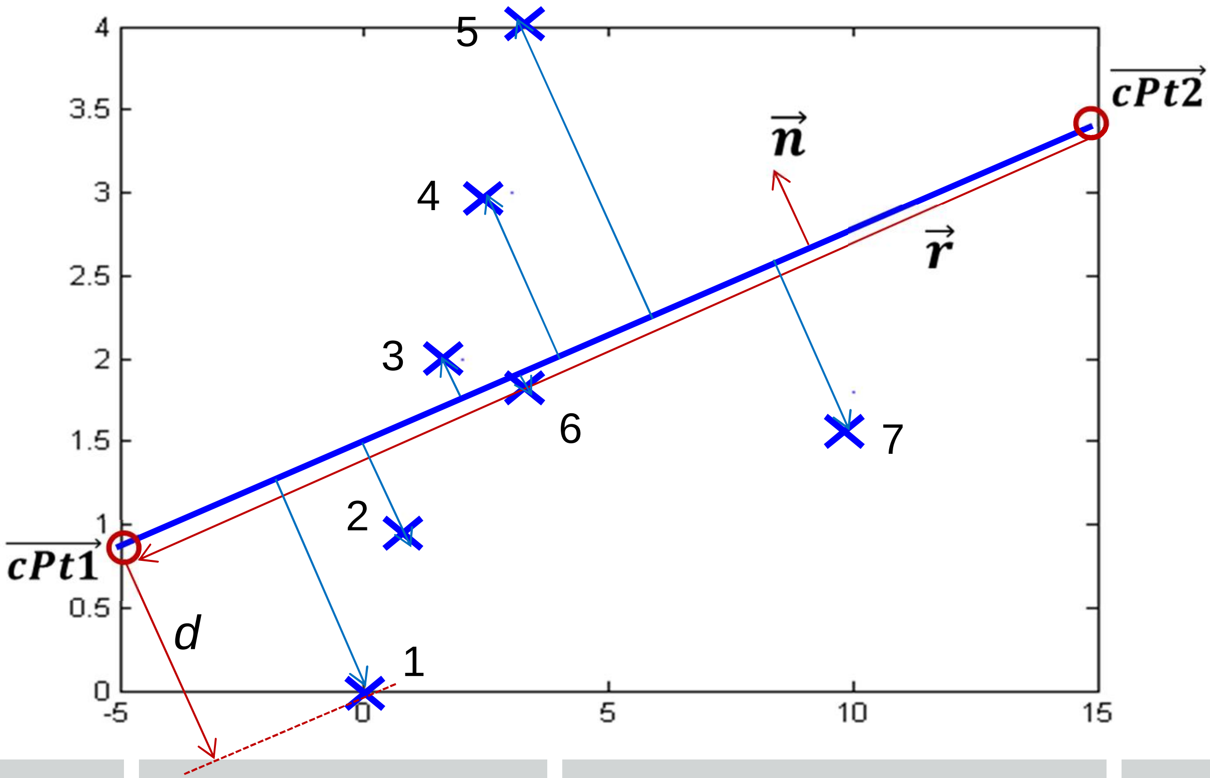
2) Calculate normal vector to direction and normalize it:

$$\vec{n} = \begin{pmatrix} -y_{\vec{r}} \\ x_{\vec{r}} \end{pmatrix}$$

$$\vec{n} = \vec{n}/\mathrm{norm}(\vec{n})$$

3) Distance to origin is given by the scalar product of some point on the line and the normal:

$$d = c\vec{P}t1^T \cdot \vec{n}$$



1) Pick two points on the line and calculate direction:

$$\vec{r} = c\vec{P}t2 - c\vec{P}t1$$

2) Calculate normal vector to direction and normalize it:

$$\vec{n} = \begin{pmatrix} -y_{\vec{r}} \\ x_{\vec{r}} \end{pmatrix}$$

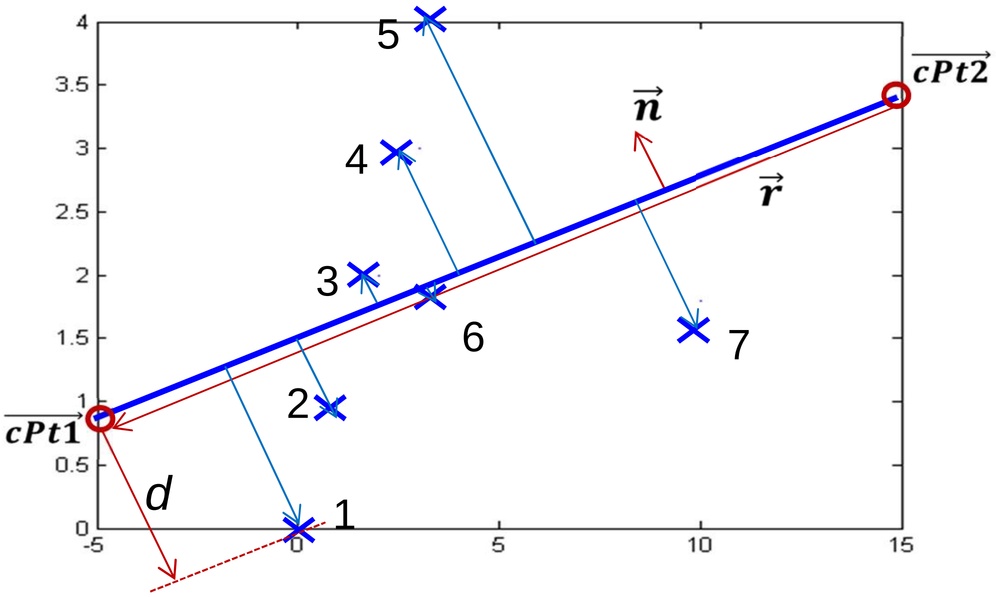
$$\vec{n} = \vec{n}/\mathrm{norm}(\vec{n})$$

3) Distance to origin is given by the scalar product of some point on the line and the normal:

$$d = c\vec{P}t1^{T} \cdot \vec{n}$$

4) Hesse normal form

$$\vec{ds} = p\vec{t}s_i^{T} \cdot \vec{n} - d$$



Implementation hints:

- How to pick two points and compute ***r***
  x = [-min(pts(:,1))-5 max(pts(:,1))+5];
  *y = m*x + t;*
  cPt1 = [x(1) y(1)];   cPt2 = [x(2) y(2)];
  r = cPt2 – cPt1;
- Use * for scalar product! Do not use loop!
- Dimension size: $\vec{n}$: 2x1 vector
  d : 1x1 scalar
  $\vec{ds}$: nx1 vector

1) Pick two points on the line and calculate direction:

$$\vec{r} = c\vec{P}t2 - c\vec{P}t1$$

2) Calculate normal vector to direction and normalize it:

$$\vec{n} = \begin{pmatrix} -y_{\vec{r}} \\ x_{\vec{r}} \end{pmatrix}$$

$$\vec{n} = \vec{n}/\mathrm{norm}(\vec{n})$$

3) Distance to origin is given by the scalar product of some point on the line and the normal:

$$d = c\vec{P}t1^T \cdot \vec{n}$$

4) Hesse normal form

$$\vec{ds} = p\vec{t}s_i^T \cdot \vec{n} - d$$

$$err = \mathrm{sum}(\vec{ds} > thr)/n$$

RANSAC

Number of iterations

Probability for an outlier

p_o

RANSAC

Number of iterations

Probability for an outlier

p_o

Probability for not having outliers in the minimum number of points required to build the model

$(1 - p_o)^{n_{mdl}}$

RANSAC

Number of iterations

Probability for an outlier

$$p_o$$

Probability for not having outliers in the minimum number of points required to build the model

$$(1 - p_o)^{n_{mdl}}$$

Probability of having at least one outlier in the minimum number of points for given iterations

$$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}}$$

RANSAC

Number of iterations

Probability for an outlier

p_o

Probability for not having outliers in the minimum number of points required to build the model

$(1 - p_o)^{n_{mdl}}$

Probability of having at least one outlier in the minimum number of points for given iterations

$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}}$

This should not be higher than a given probability

$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}} \leq 1 - P_{corr}$

RANSAC

Number of iterations

Probability for an outlier

$$p_o$$

Probability for not having outliers in the minimum number of points required to build the model

$$(1-p_o)^{n_{mdl}}$$

Probability of having at least one outlier in the minimum number of points for given iterations

$$(1-(1-p_o)^{n_{mdl}})^{n_{it}}$$

This should not be higher than a given probability

$$(1-(1-p_o)^{n_{mdl}})^{n_{it}} \leq 1 - P_{corr}$$

$$\Rightarrow n_{it} = \left\lceil \frac{log(1-P_{corr})}{log(1-(1-p_o)^{n_{mdl}})} \right\rceil$$

RANSAC

Number of iterations

Probability for an outlier

$$p_o$$

Probability for not having outliers in the minimum number of points required to build the model

$$(1 - p_o)^{n_{mdl}}$$

Probability of having at least one outlier in the minimum number of points for given iterations

$$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}}$$

This should not be higher than a given probability

$$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}} \leq 1 - P_{corr}$$

$$\Rightarrow n_{it} = \left\lceil \frac{log(1 - P_{corr})}{log(1 - (1 - p_o)^{n_{mdl}})} \right\rceil$$

Estimate probability for an outlier using relative frequencies. Minimum number of points for the model is given.

→ Choose probability for having at least one iteration without outliers

RANSAC

Task: `commonransac:` In `it` iterations choose randomly `mn` points out of `data`. Use them to estimate the model with `mdlEstFct`. Estimate the error for this model using `errFct`.

For each iteration, do

1. Randomly choose `mn` points from data
   → could use `randperm()`
2. Use them to estimate the model with `mdlEstFct()`
3. Compute the error for this model using `errFct()`