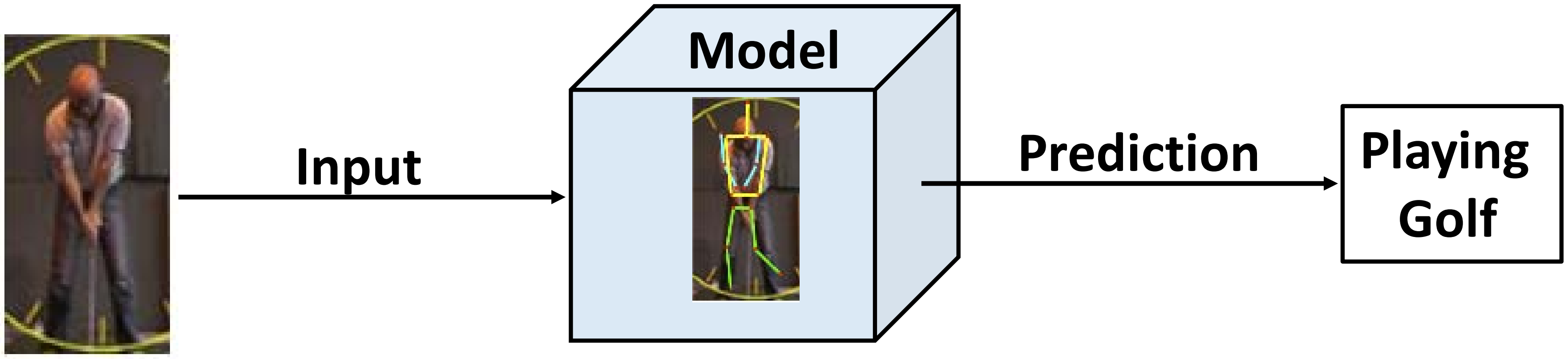
CSE 473/573
RANSAC & Least Squares

Devansh Arpit

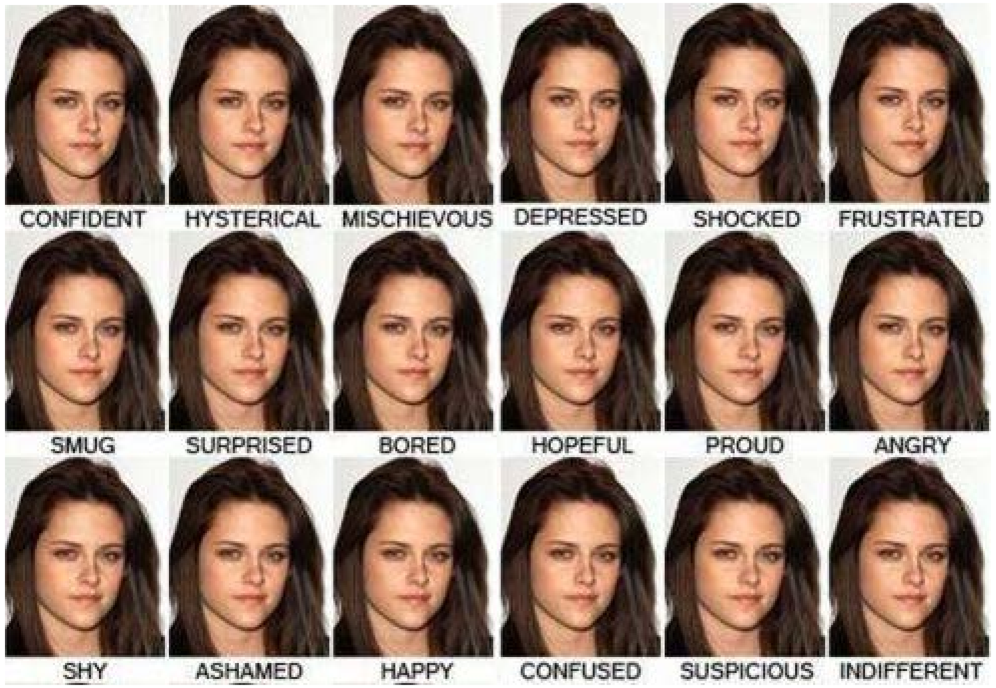
Mathematical Models

- Compact Understanding of the World



Mathematical Models - Example

- Face Recognition with varying expressions

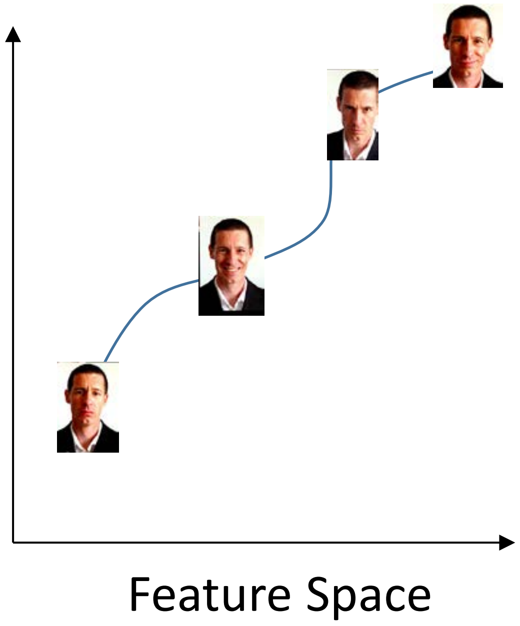


- Too Easy…

Mathematical Models

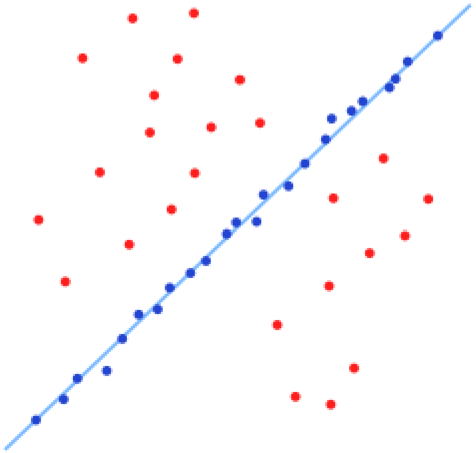
- Face Recognition with varying expressions

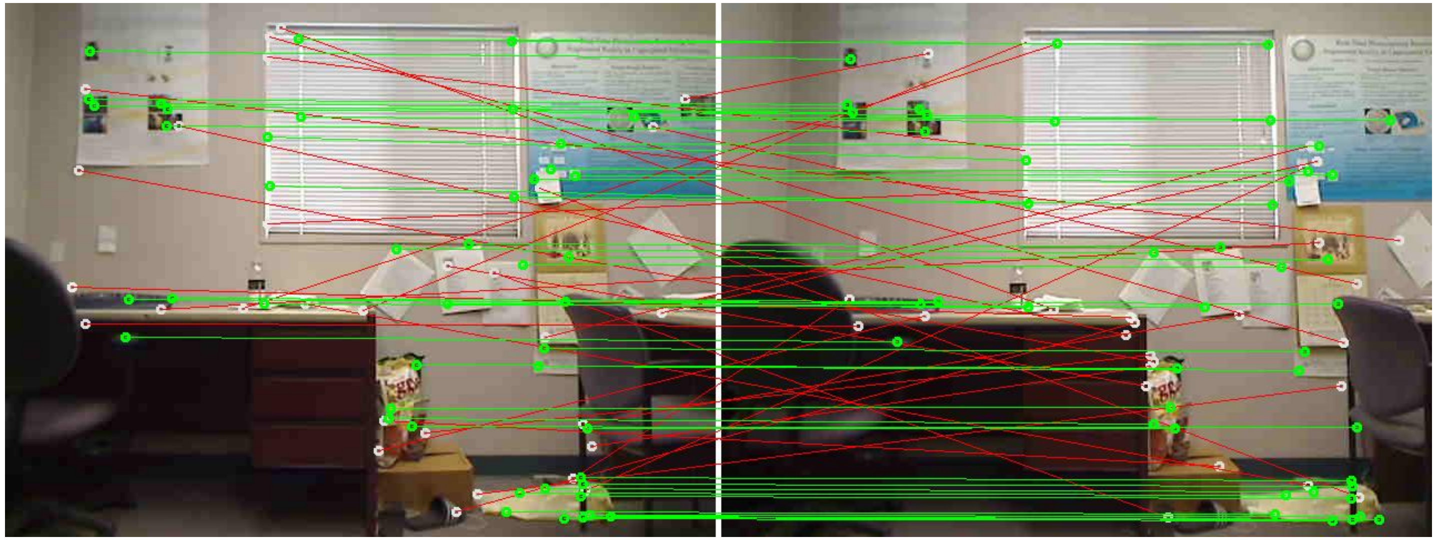
Learn Model

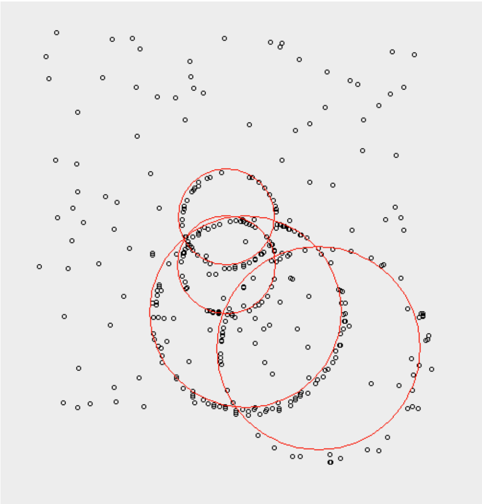


I. RANSAC

- Random Sample Consensus
- Used for Parametric Matching/Model Fitting
- Applications:

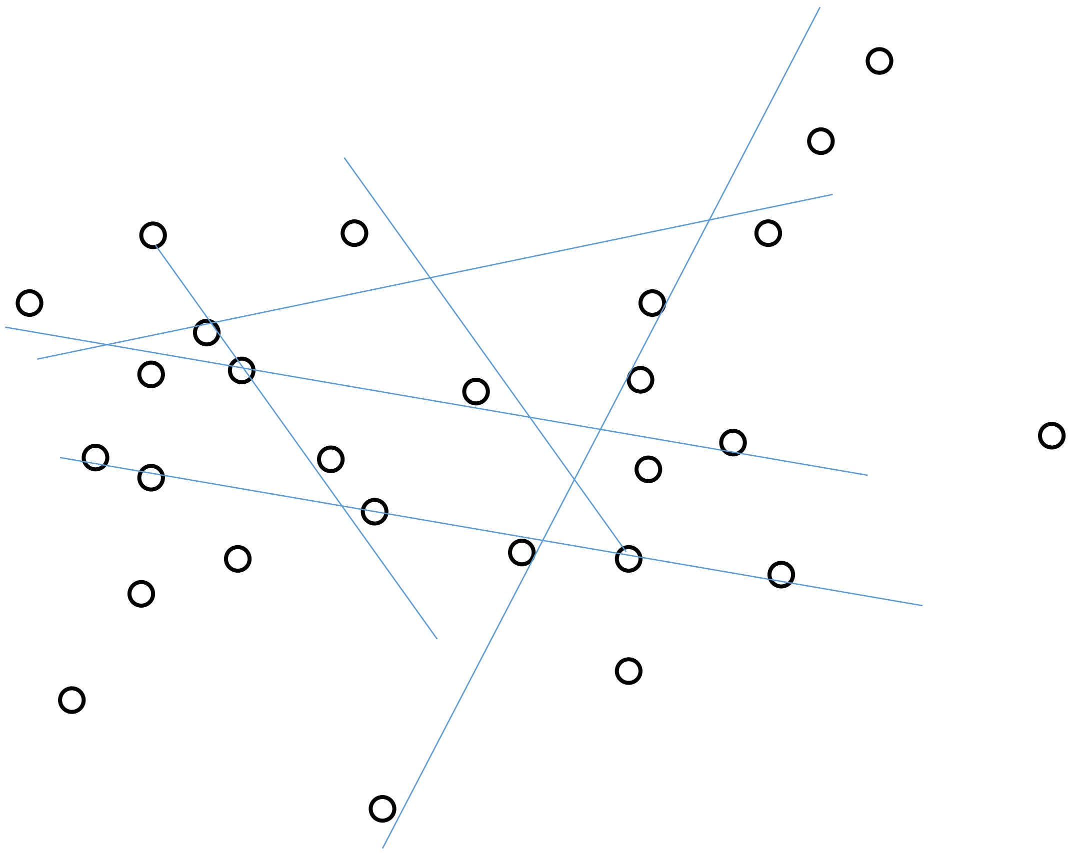






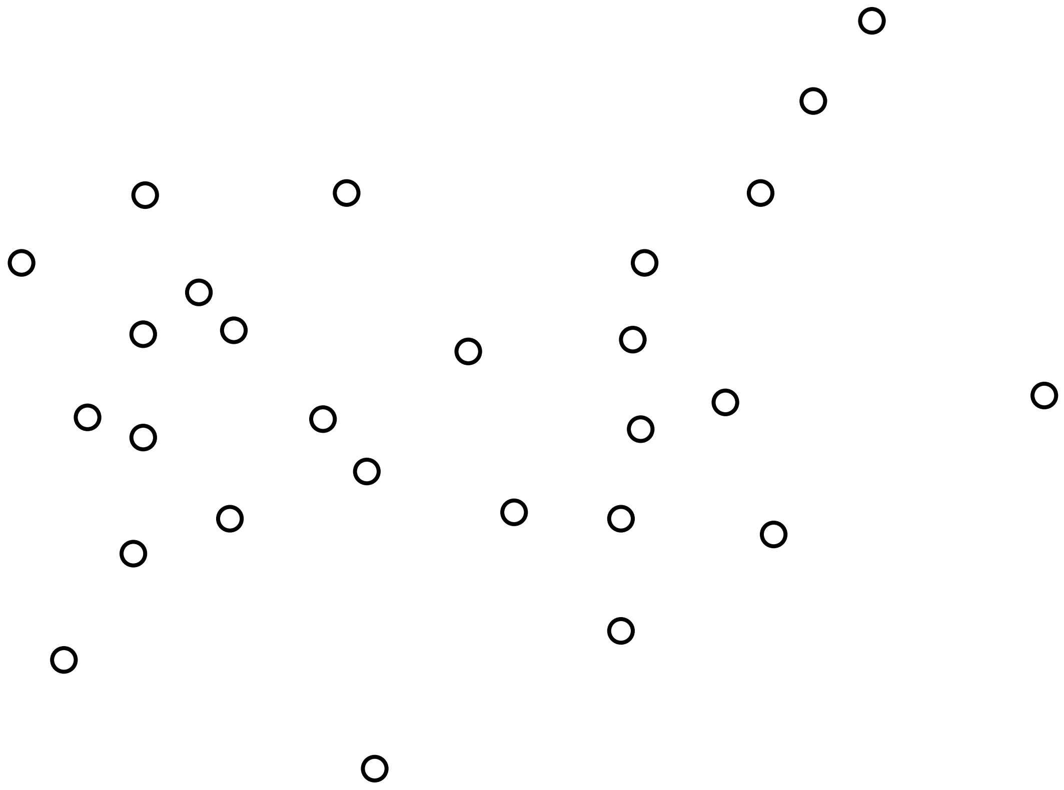
Line Fitting

- Fit the best possible Line to these points
- Brute Force Search – 2^N possibilities!!!
- Not Feasible
- Better Strategy?



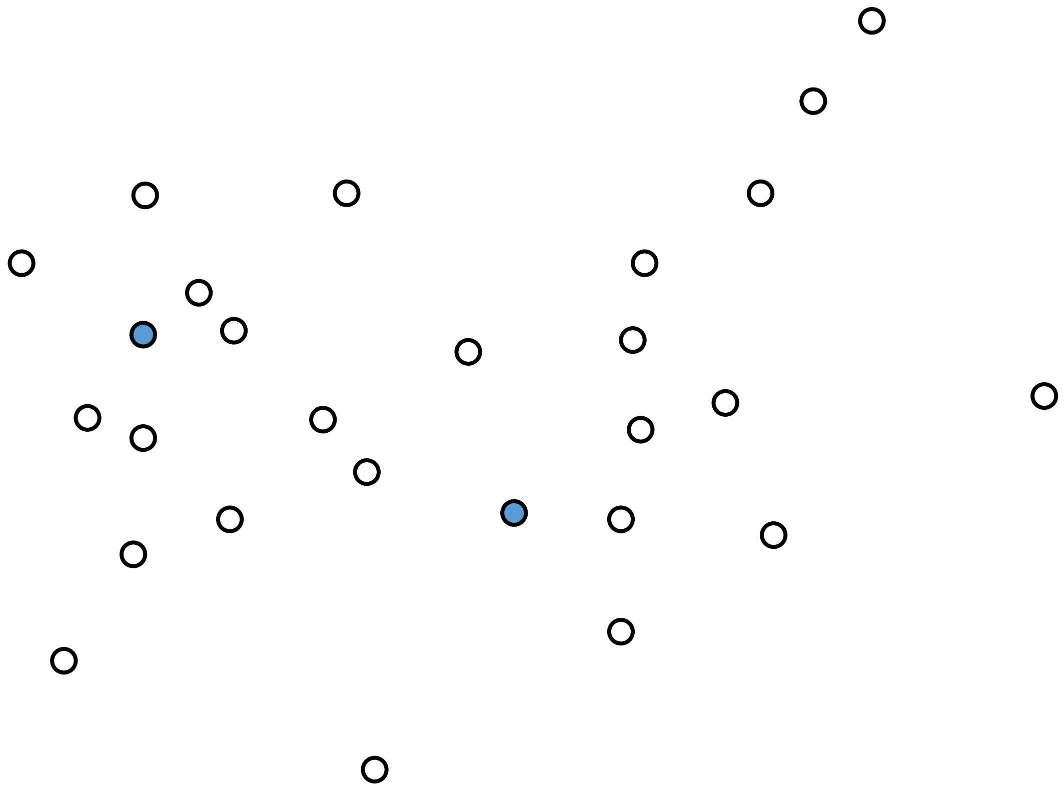
How RANSAC Works

- Random Search – Much Faster!!!



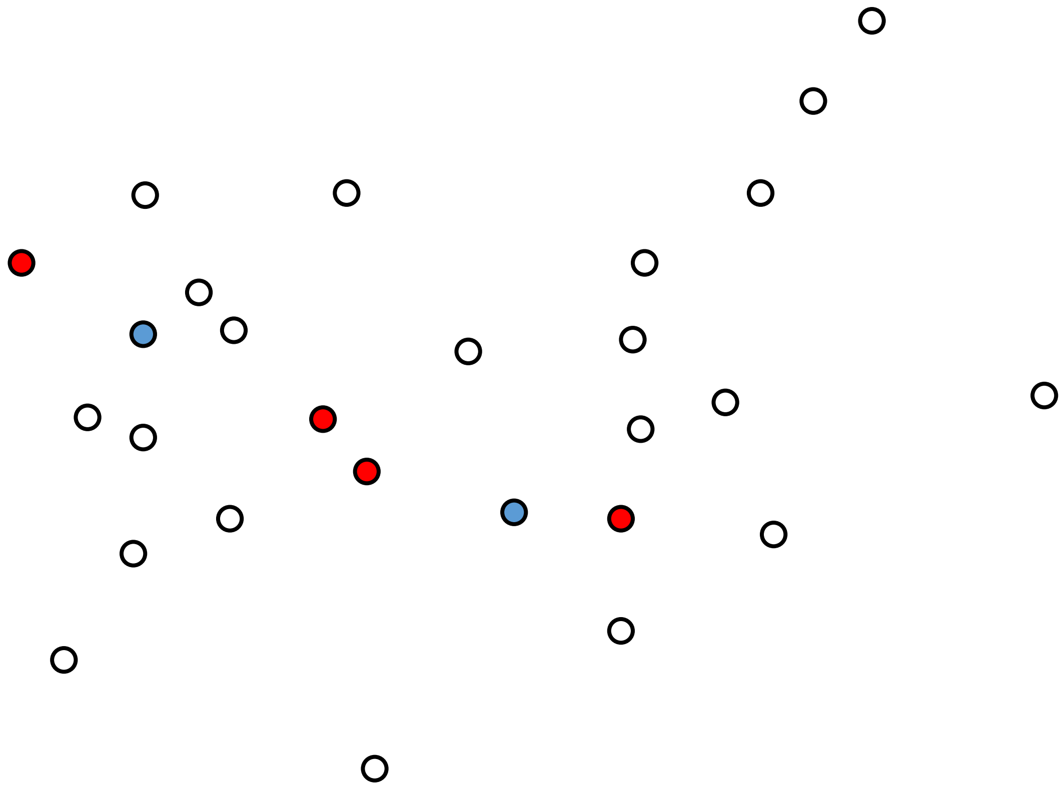
Line Fitting using RANSAC

- Iteration 1



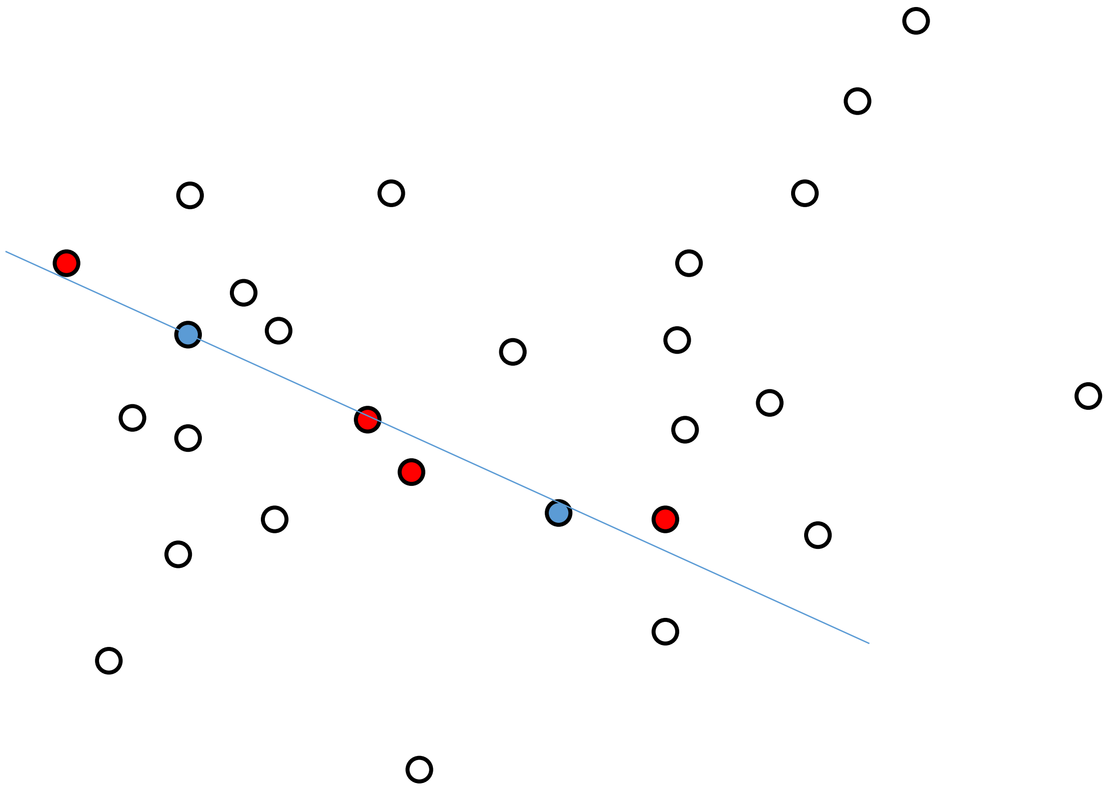
Line Fitting using RANSAC

- Iteration 1



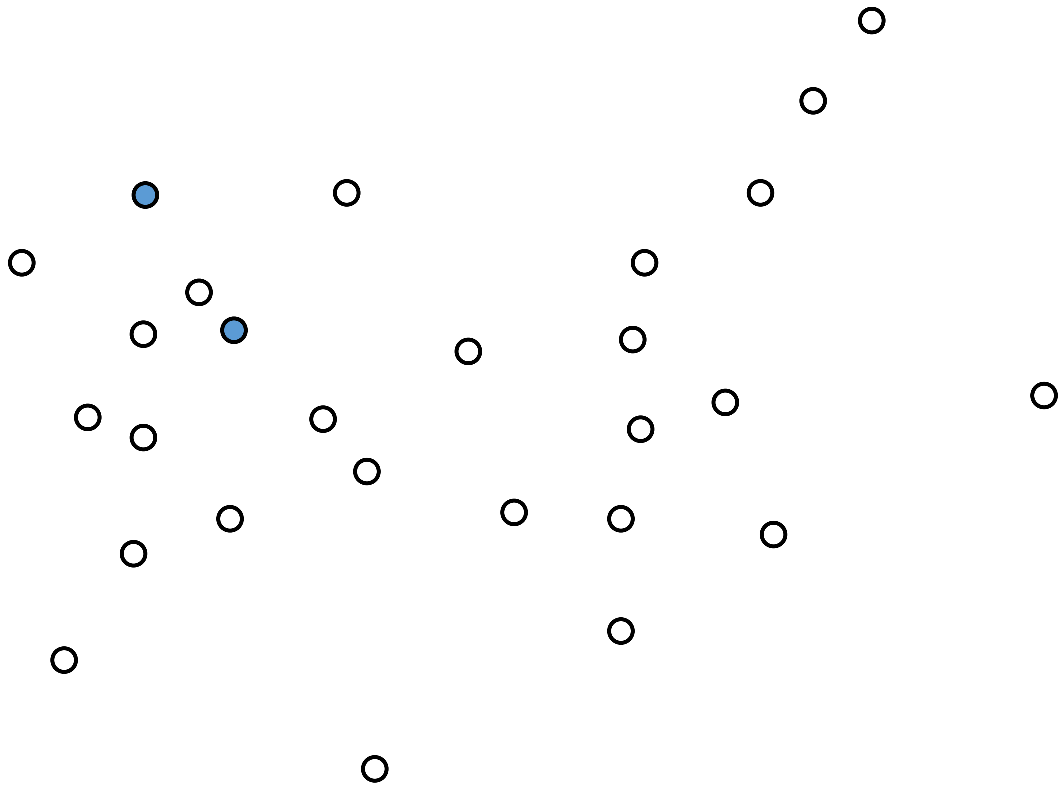
Line Fitting using RANSAC

- Iteration 1



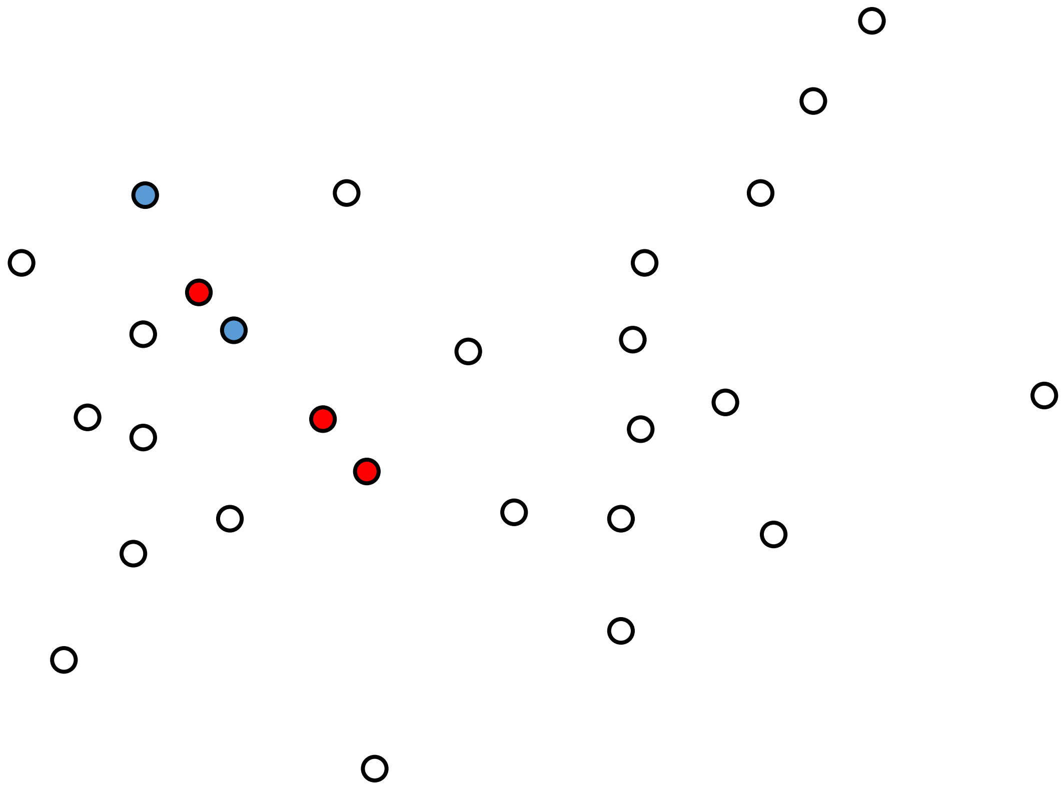
Line Fitting using RANSAC

- Iteration 2



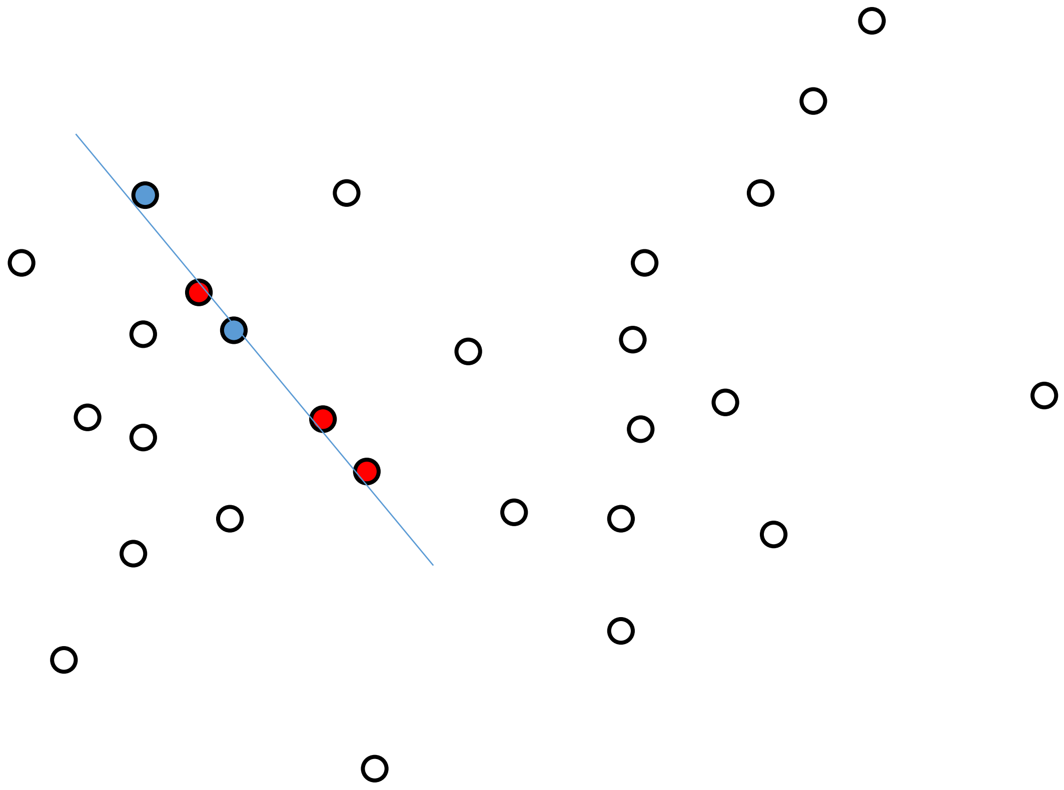
Line Fitting using RANSAC

- Iteration 2



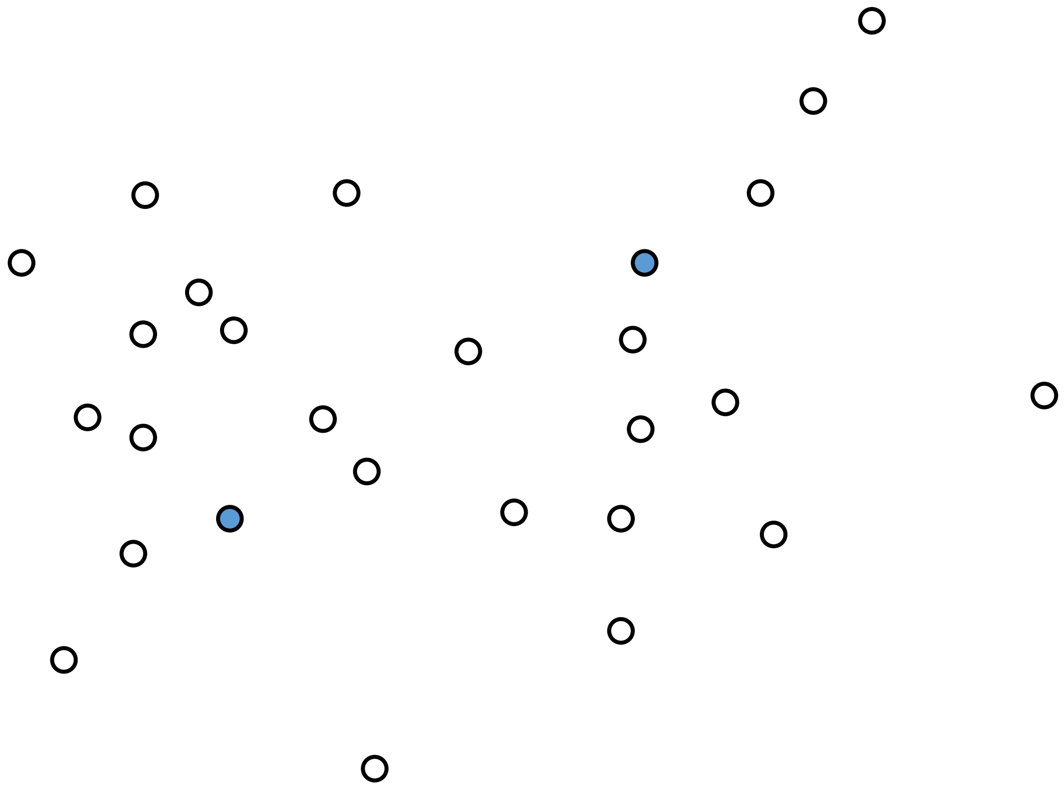
Line Fitting using RANSAC

- Iteration 2



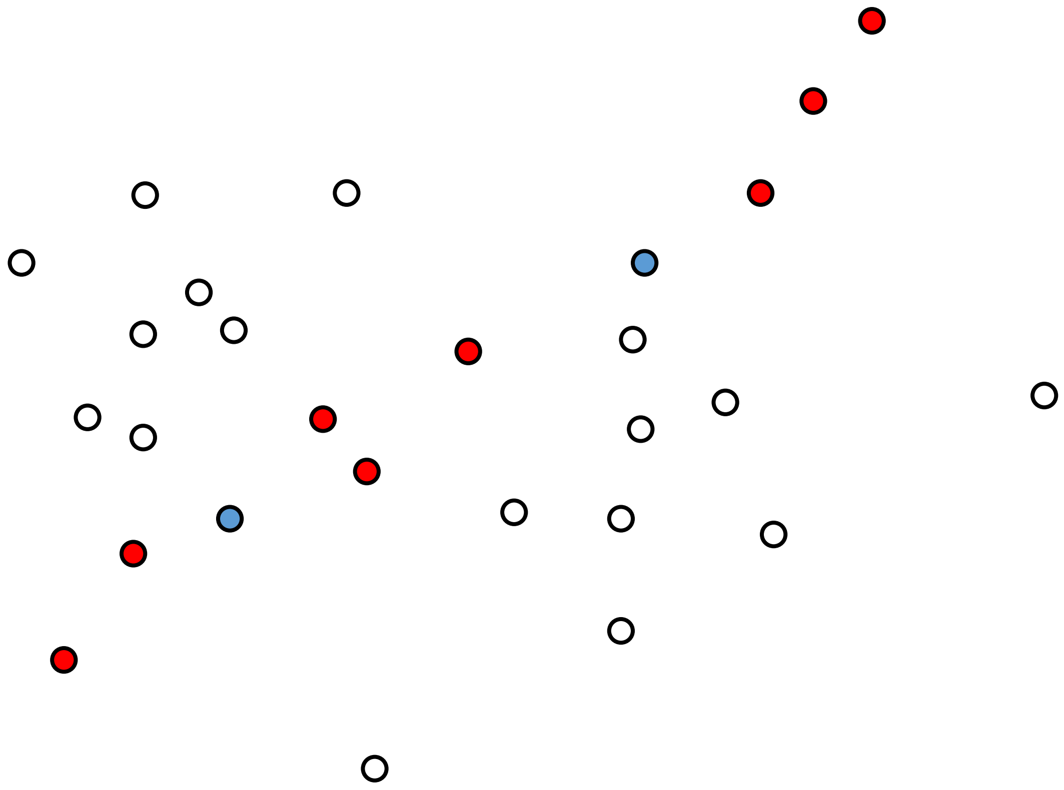
Line Fitting using RANSAC

- …
- Iteration 5



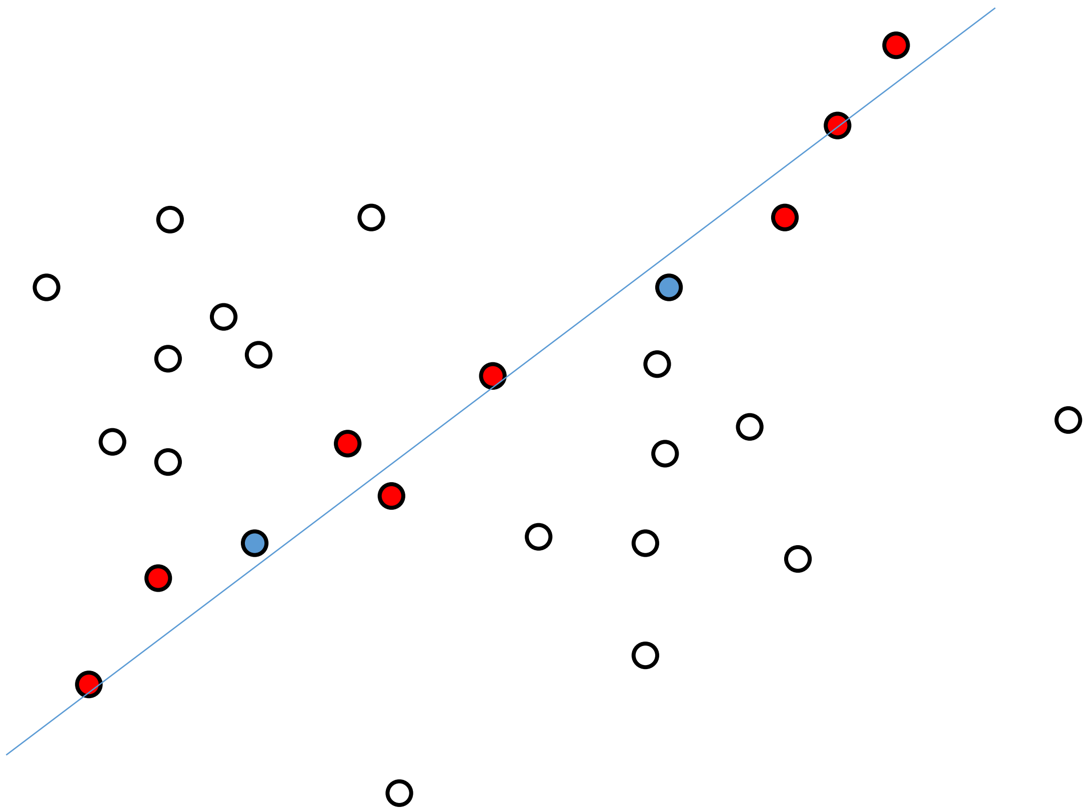
Line Fitting using RANSAC

- Iteration 5



Line Fitting using RANSAC

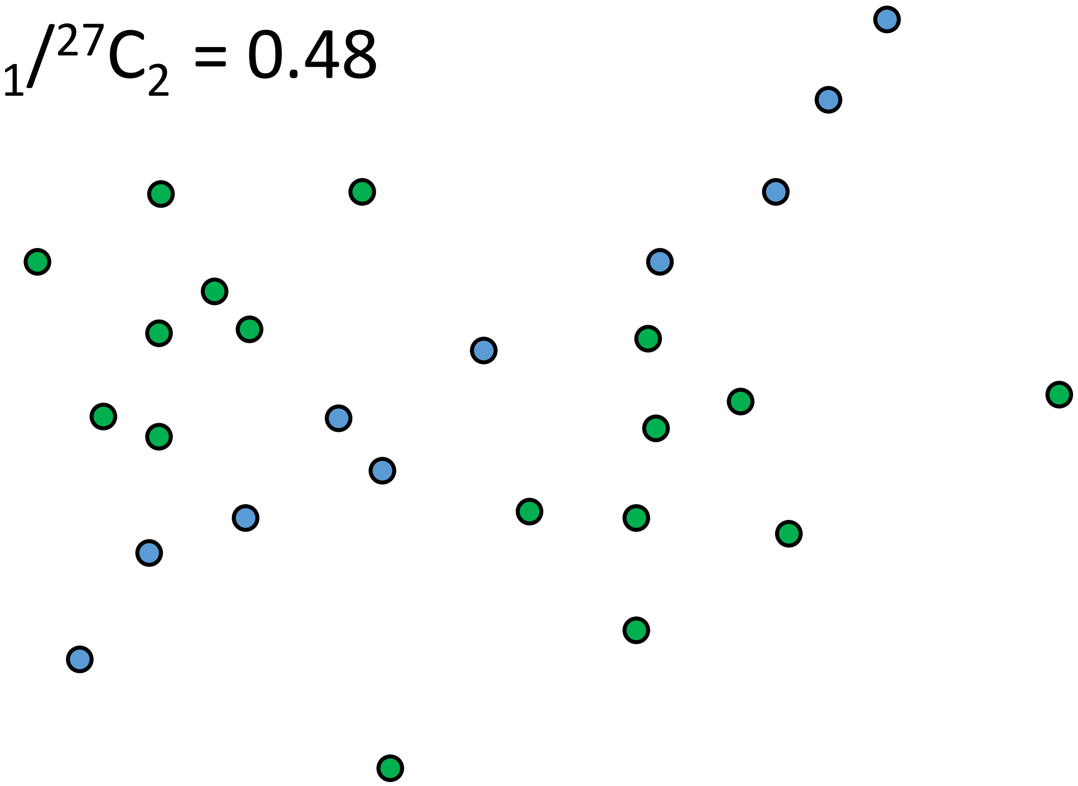
- Iteration 5



Why RANSAC Works?

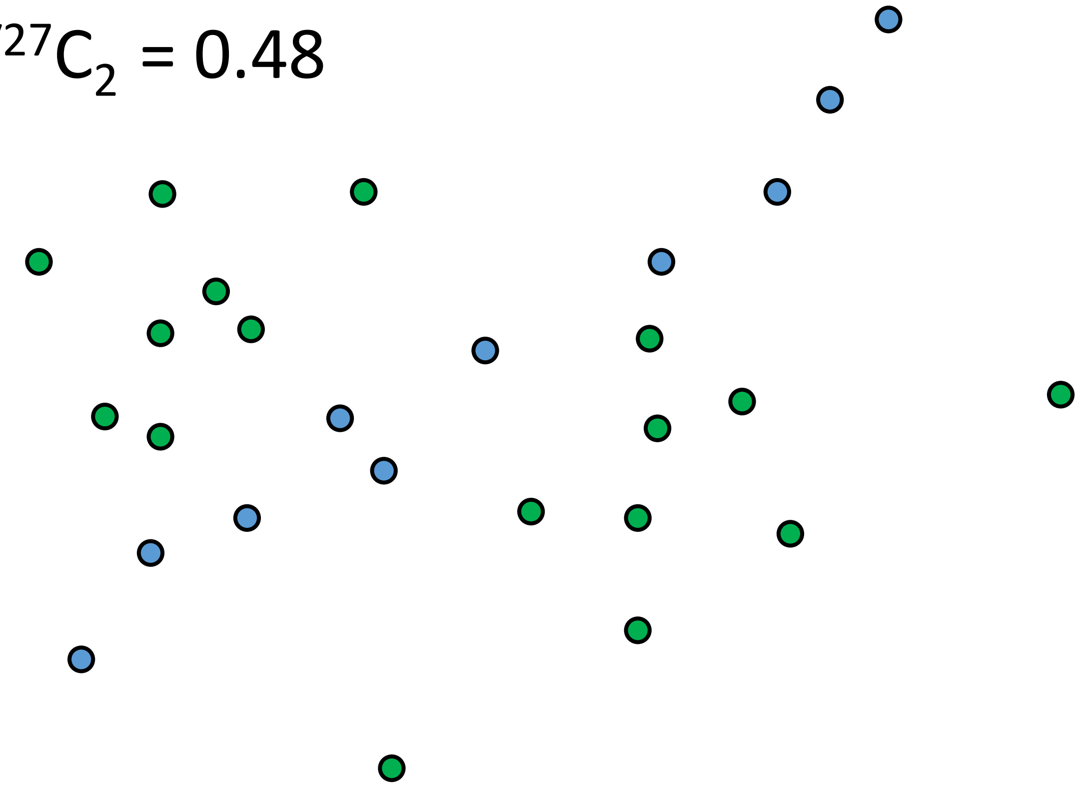
- Inliers vs Outliers

- P(selecting outliers) = $^{17}C_2/^{27}C_2 + {^{17}C_1}{^{10}C_1}/^{27}C_2 = 0.48$



Why RANSAC Works?

- Inliers vs Outliers
- P(selecting outliers) $^{17}C_2/^{27}C_2 + {}^{17}C_1{}^{10}C_1/^{27}C_2 = 0.48$
- After 5 iterations…
- P(selecting outliers) = $(0.48)^5 = 0.026$



Why RANSAC Works?

- In general:

- $p = 1 – (1 - w^n)^k$

Where,
p = probability for selecting inliers
w = ratio of inliers to total #points
n = minimum #points required (for line = 2, circle =3)
k = #iterations

RANSAC Algorithms

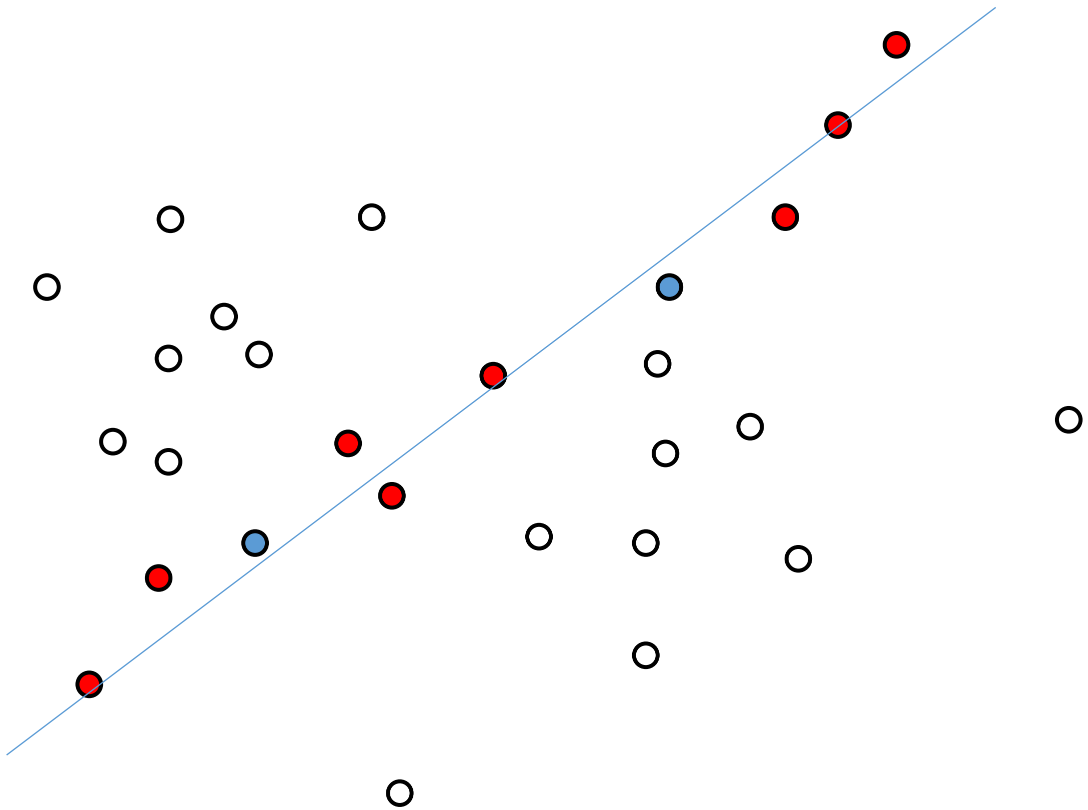
Determine:
- n—the smallest number of points required (e.g., for lines, $n = 2$, for circles, $n = 3$)
- k—the number of iterations required
- t—the threshold used to identify a point that fits well
- d—the number of nearby points required to assert a model fits well

Until k iterations have occurred
- Draw a sample of n points from the data uniformly and at random
- Fit to that set of n points
- For each data point outside the sample
  - Test the distance from the point to the structure against t; if the distance from the point to the structure is less than t, the point is close
- end
- If there are d or more points close to the structure then there is a good fit. Refit the structure using all these points. Add the result to a collection of good fits.

end

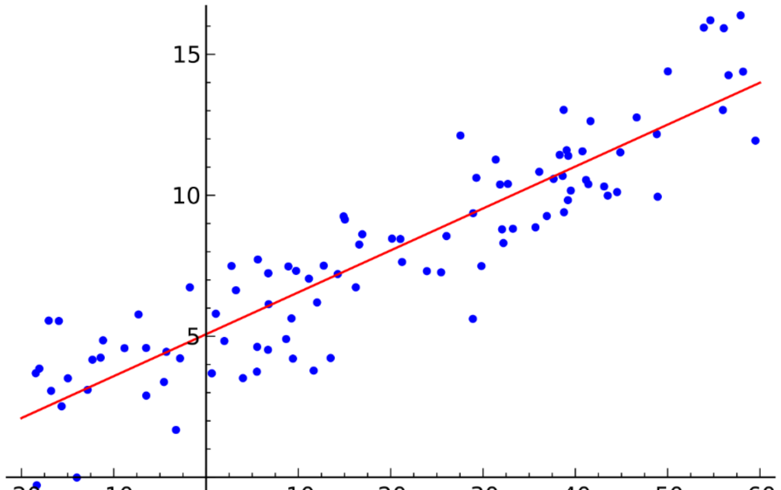
Use the best fit from this collection, using the fitting error as a criterion

Algorithm 10.4: RANSAC: Fitting Structures Using Random Sample Consensus.

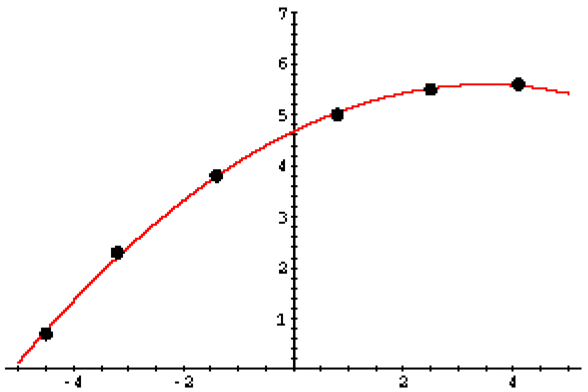


II. Least Squares

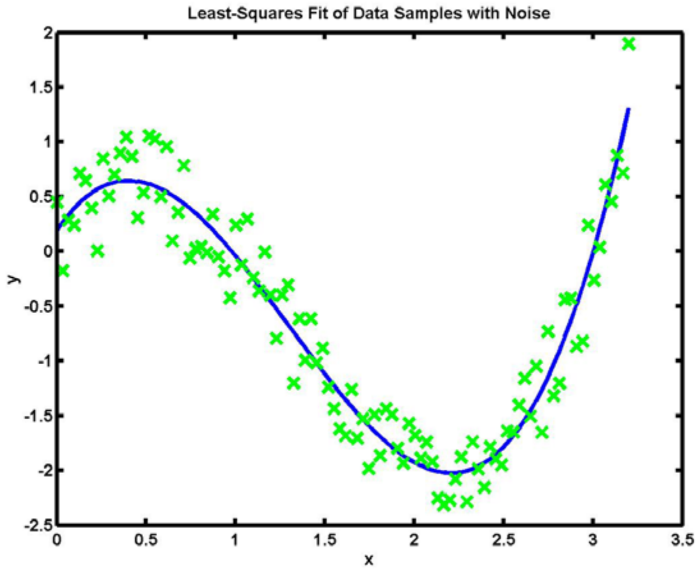
- Fitting Curves/Learning Data Manifolds



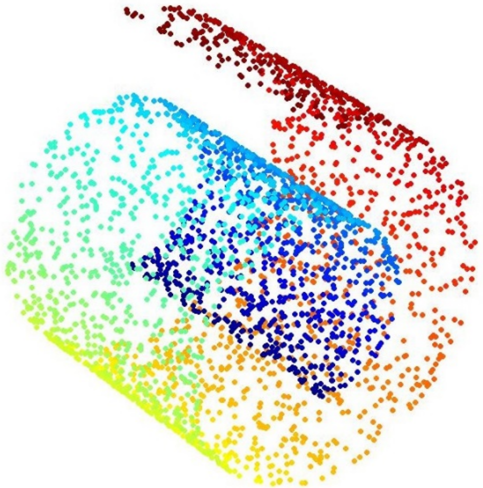
Fitting Line



Fitting Quadratic Curve



Fitting Higher Degree Polynomials



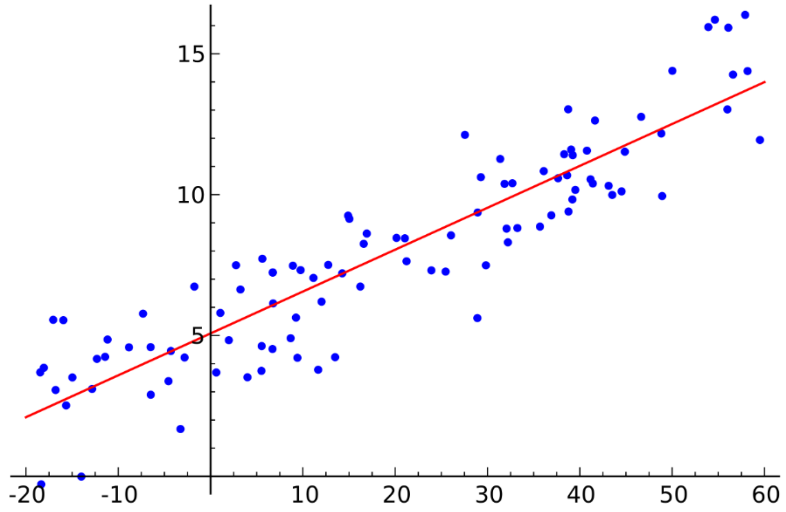
Learning Manifolds

Line Fitting

- Goal: Find a line that best explains the observed data

- Target: y_i
- Data: x_i
- Line parameter: w,b
- Line Model:

$y_i = w\,x_i + b$



Fitting Line

Line Fitting

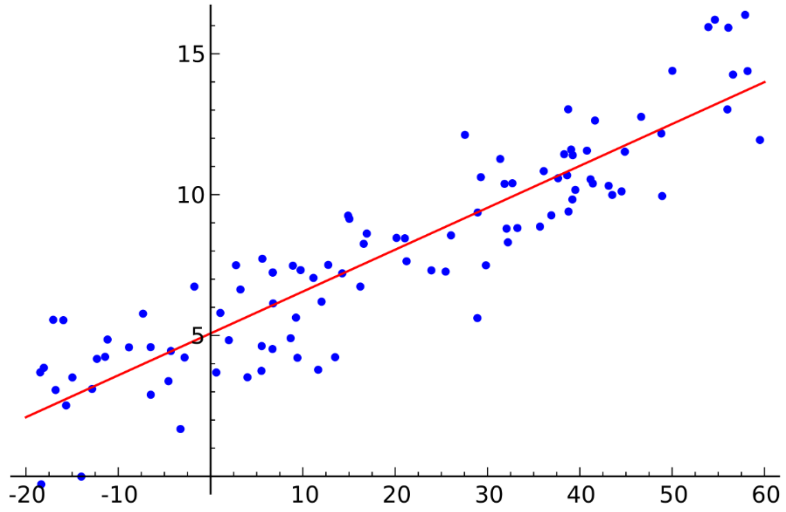
- Line Model:

$$y_i = w\,x_i + b$$

- Too many samples!

- Minimize error:

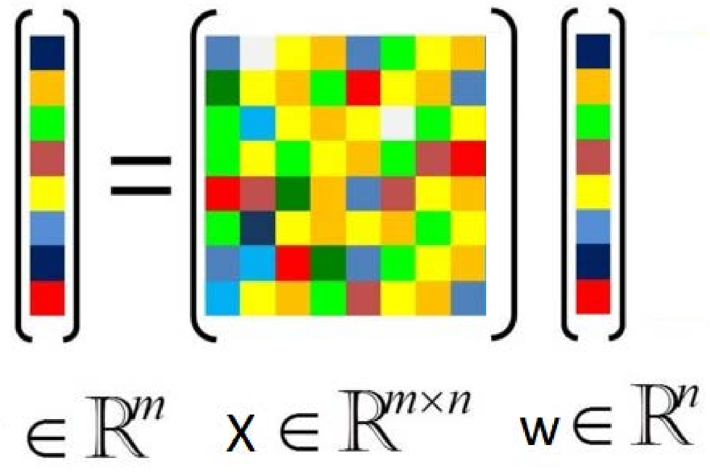
$$\min_{w,b}.\ \sum_{i=1}^{N}(y_i - w\,x_i + b)^2$$

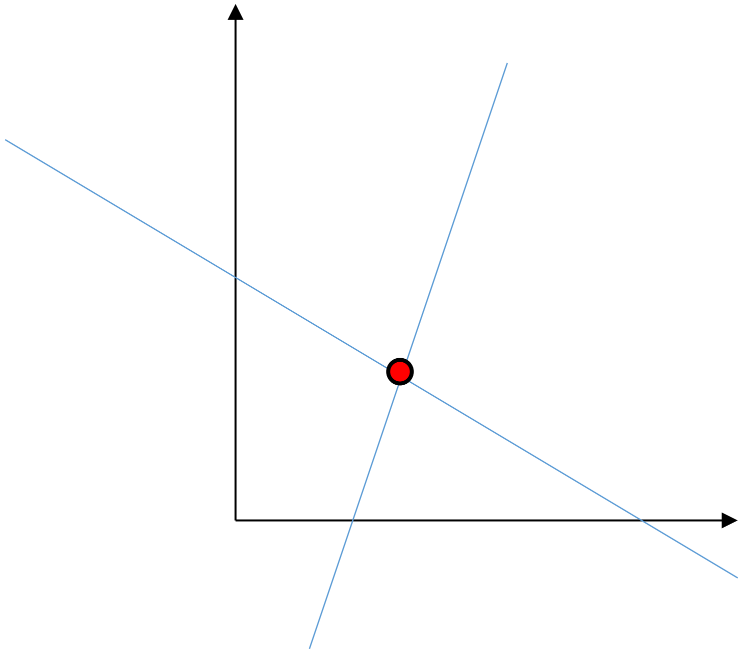


Fitting Line

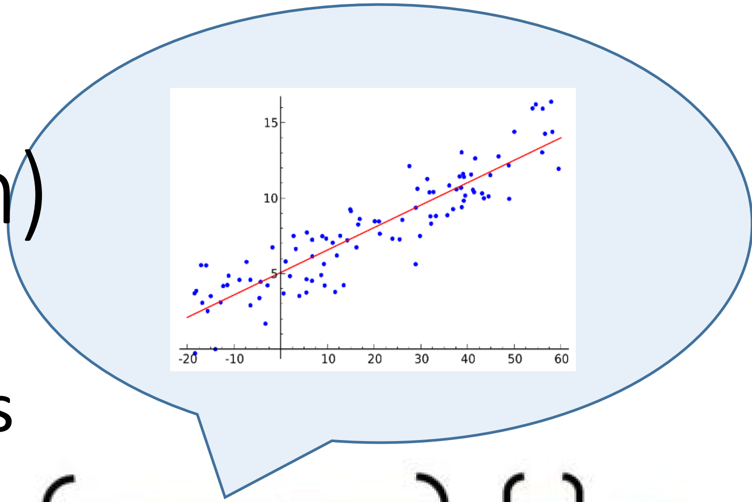
#Samples(m) vs #Model-Parameters(n)

- Case 1 (m=n): Unique Solution
- w=X\y
- No least square requires

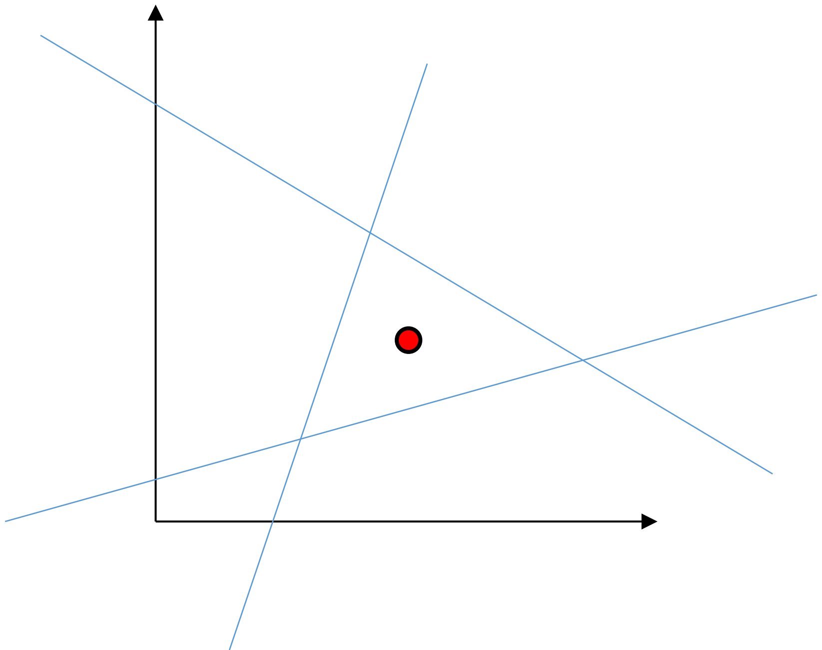


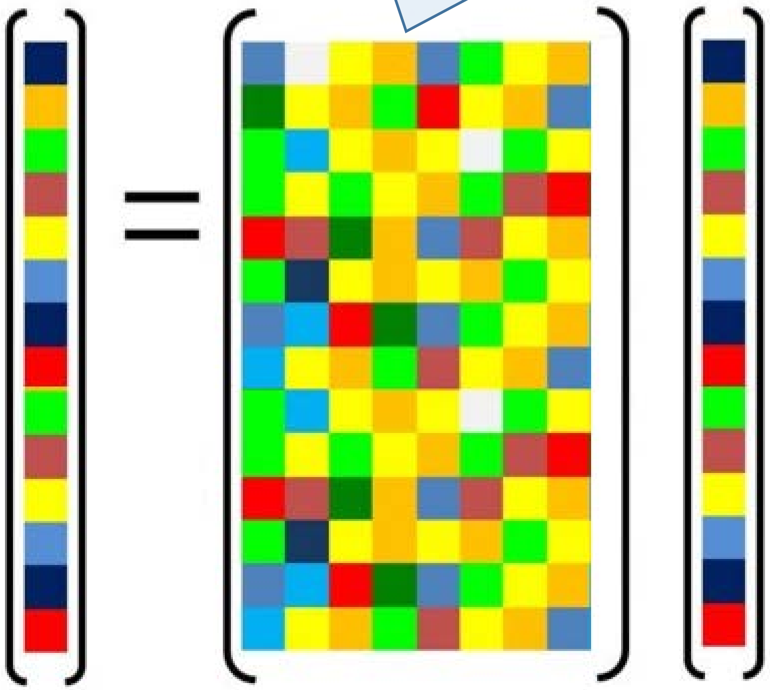


#Samples(m) vs #Model-Parameters(n)



- Case 2 (m>n): Over-determined system of equations
- No Solution exists!
- Hence, we minimize error (fitting)





$y \in \mathbb{R}^m$ $x \in \mathbb{R}^{m \times n}$ $w \in \mathbb{R}^n$

#Samples(m) vs #Model-Parameters(n)

- Case 3 (m<n): Under-determined system of equations
- Infinite Solutions exist!
- Which one to choose?