

# 04 - Normal Estimation

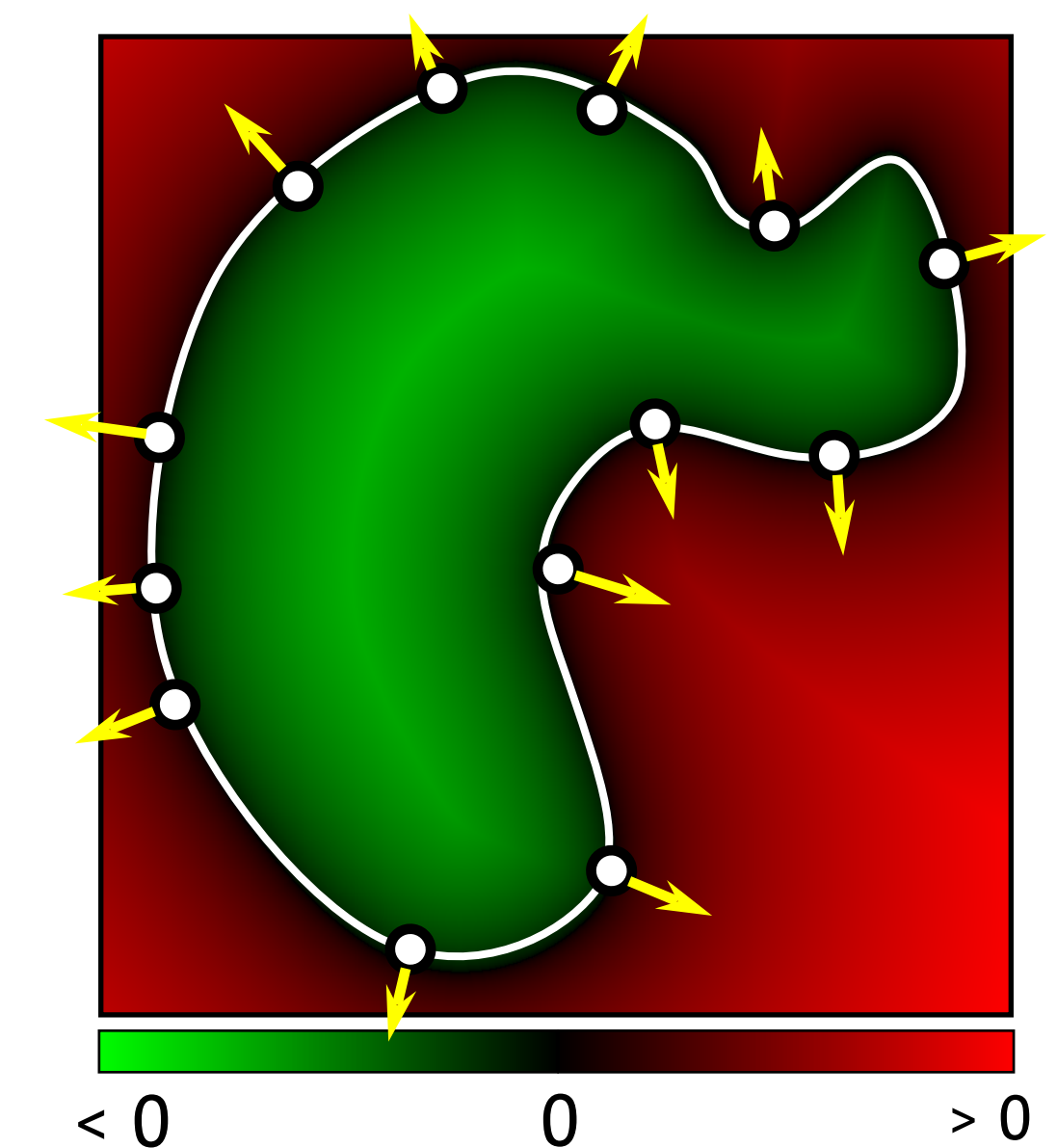
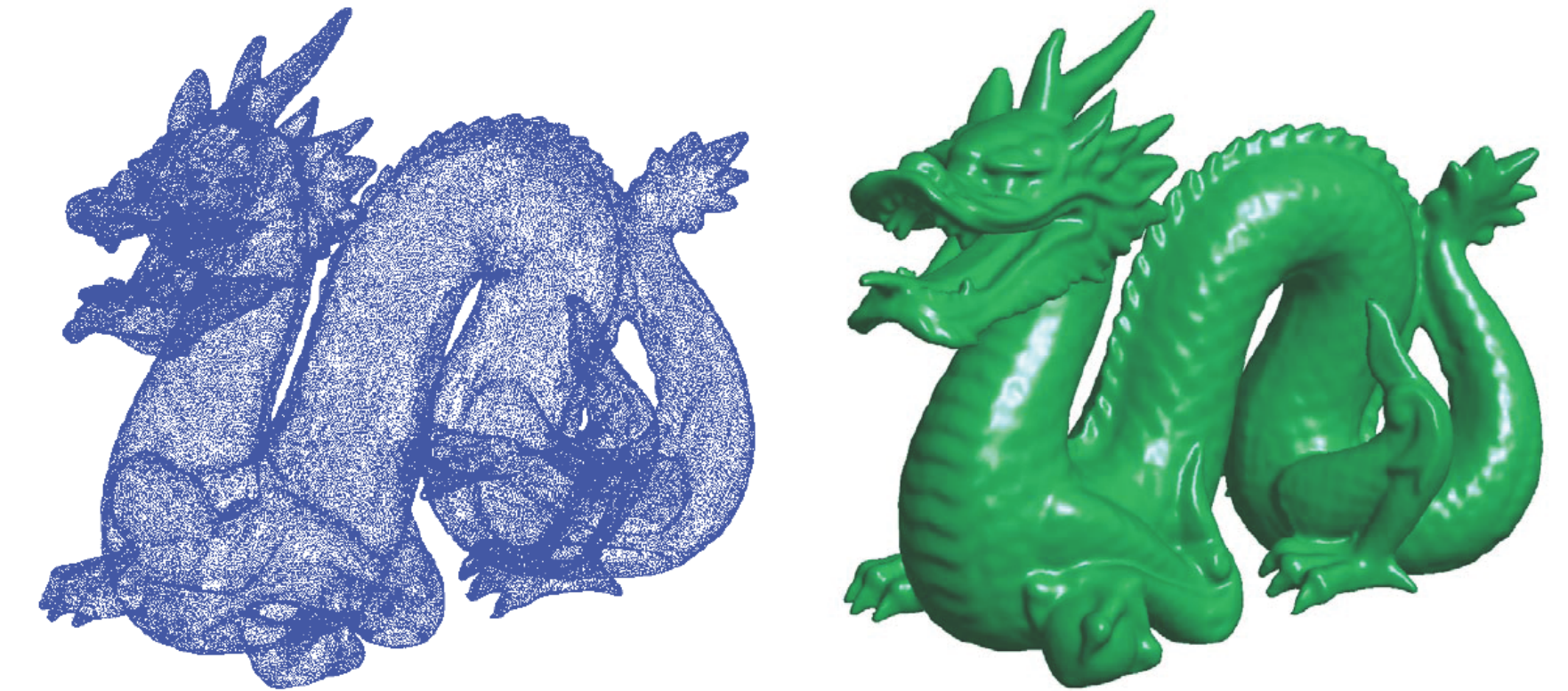
Acknowledgements: Daniele Panozzo, Olga Sorkine Hornung

# In this lecture

- What we can do with just point data
- Least squares approximation

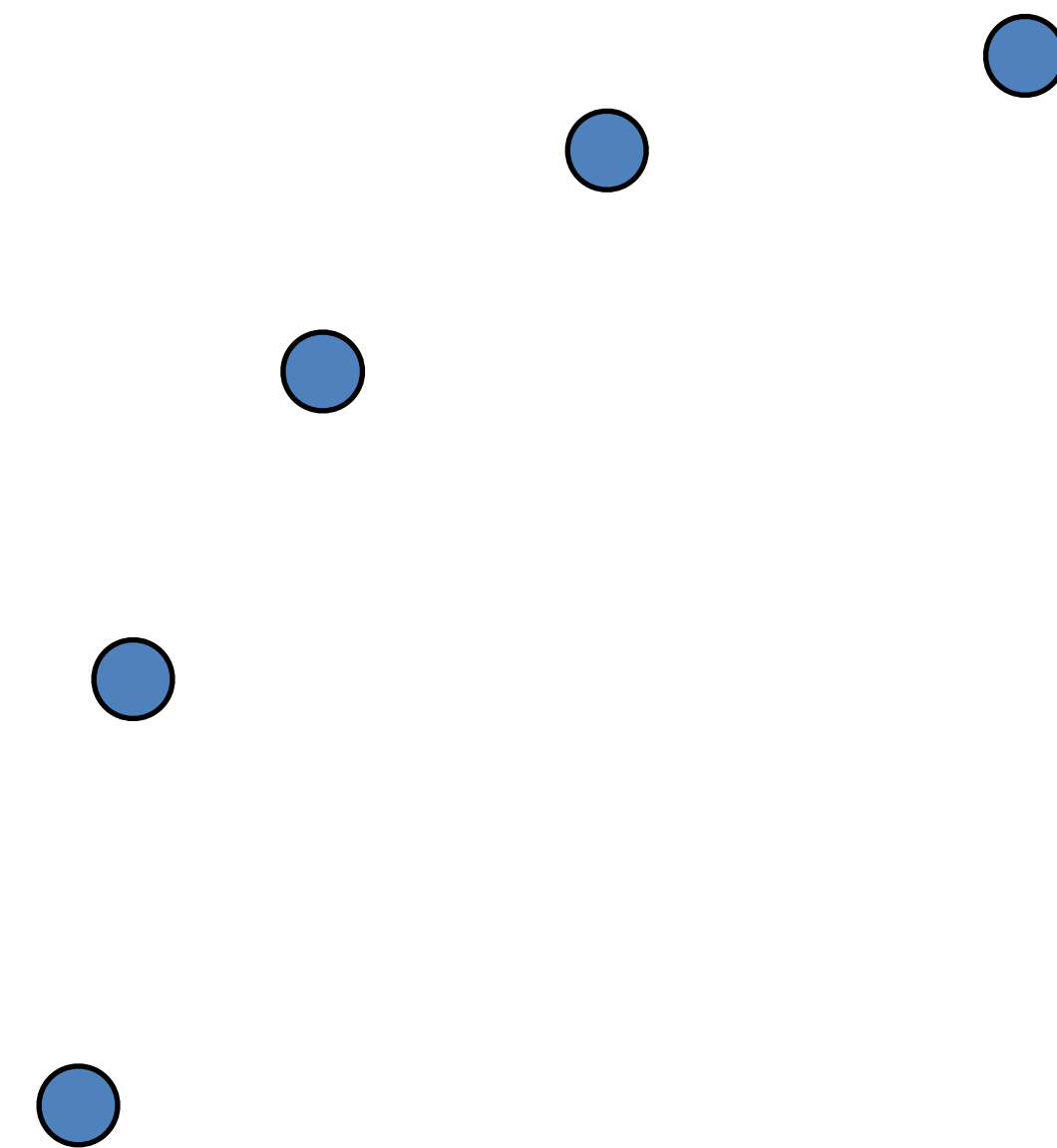
# Implicit Surface Reconstruction

- Implicit function from point clouds
  - Need to tell inside from outside
  - Boundary goes through points in the cloud
- Each point is seen as a differential of surface
- Need consistently oriented normals
- What happens between points?



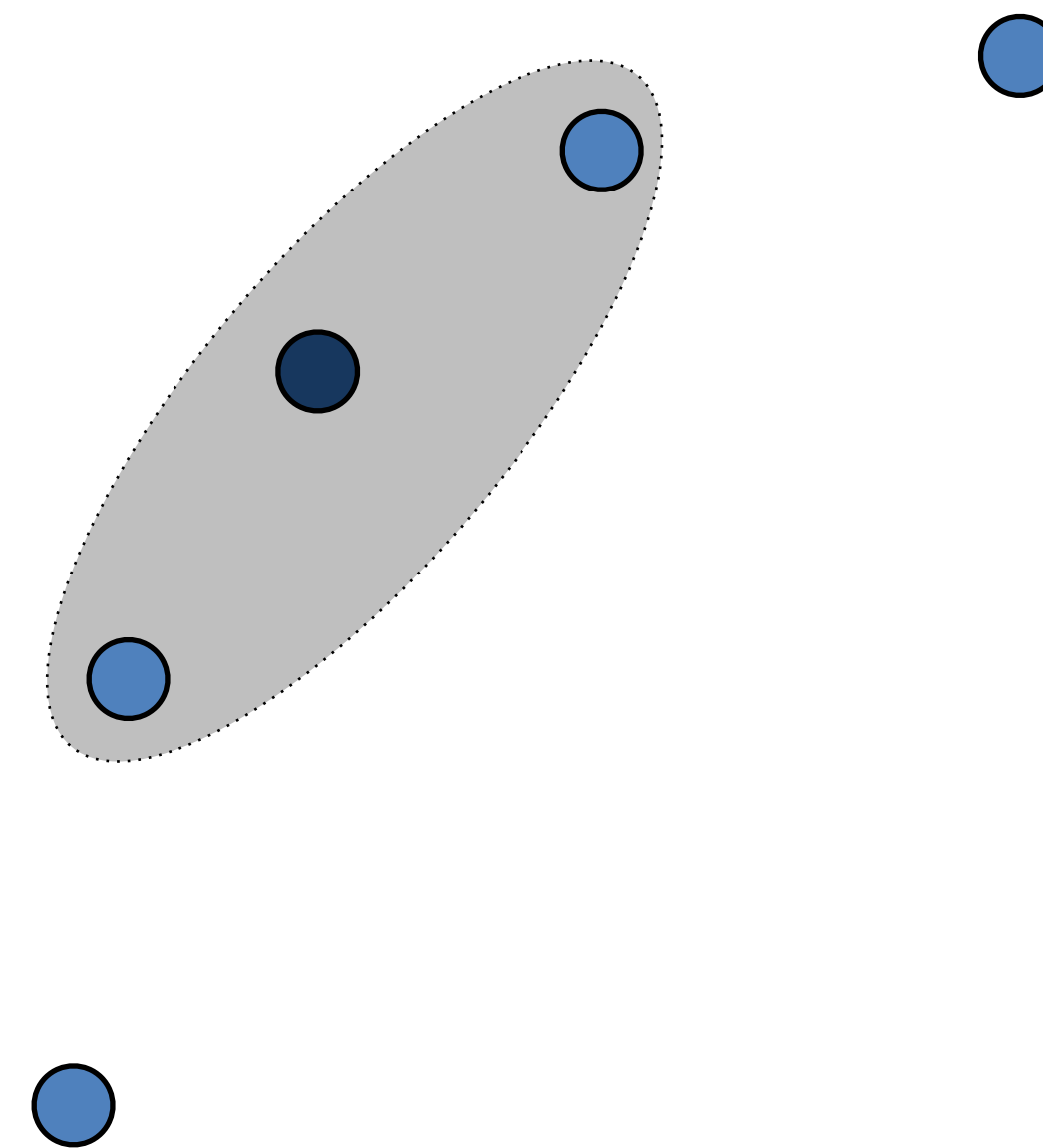
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



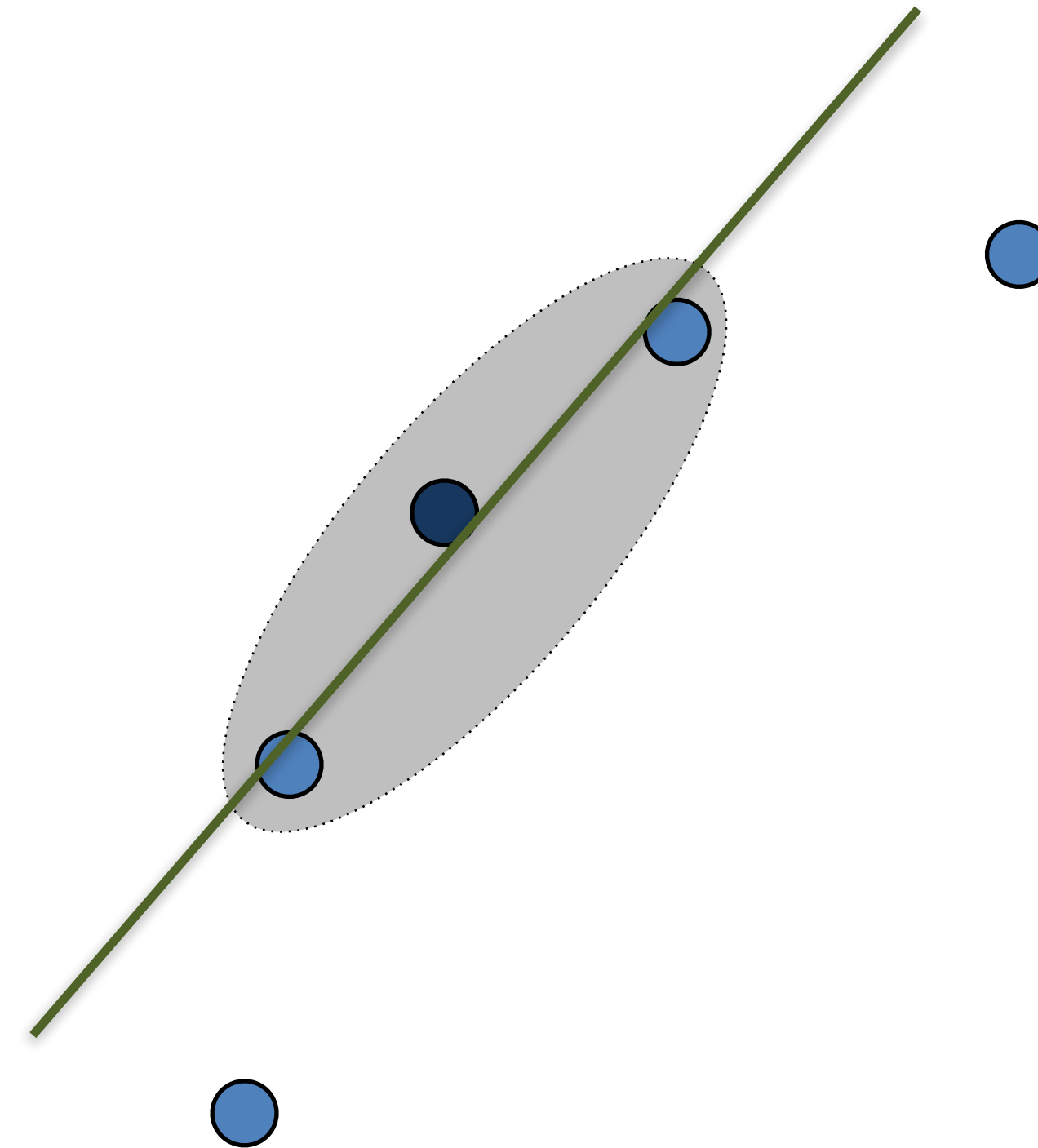
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



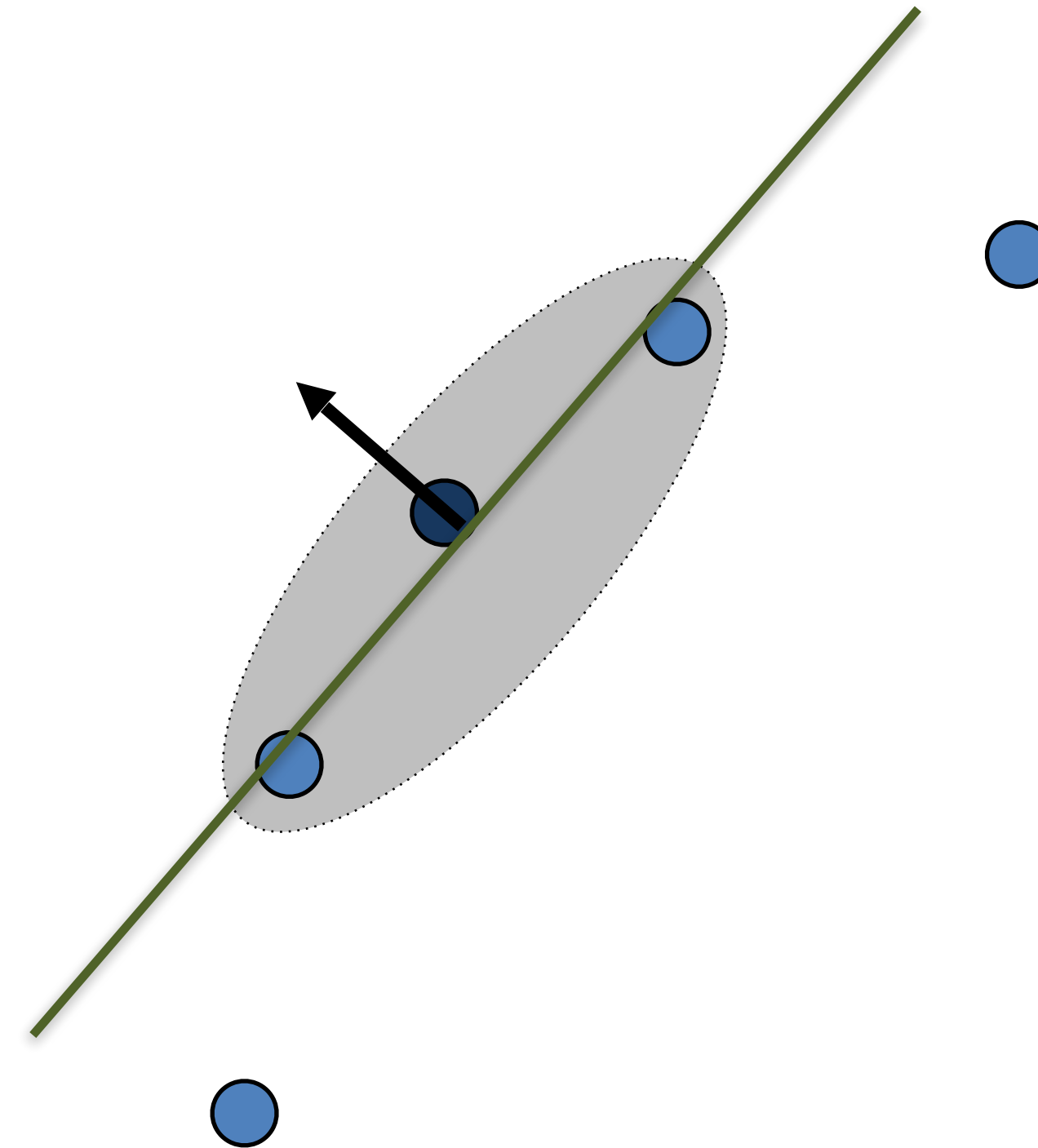
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



# Normal Estimation

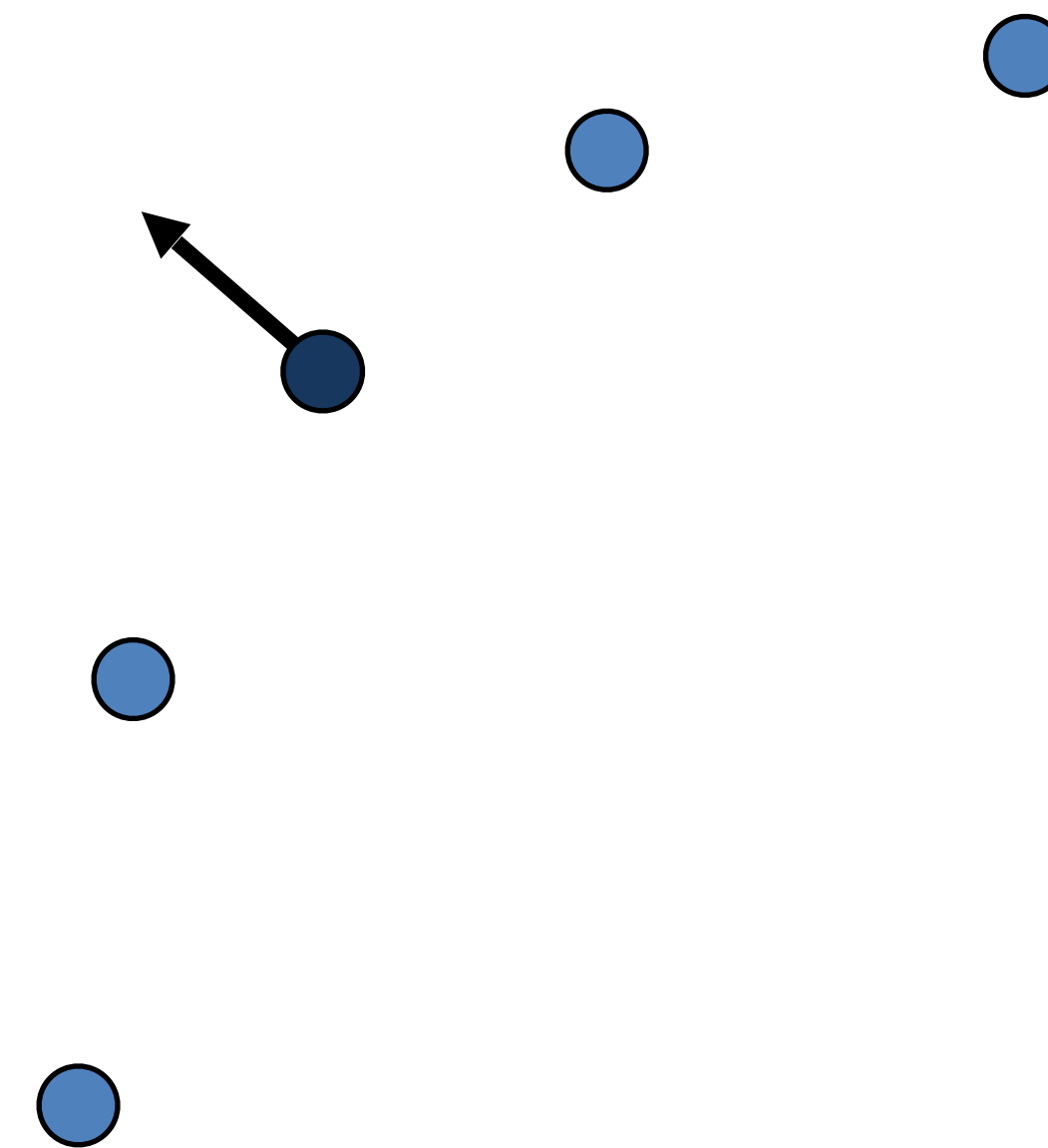
- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)





# Normal Estimation

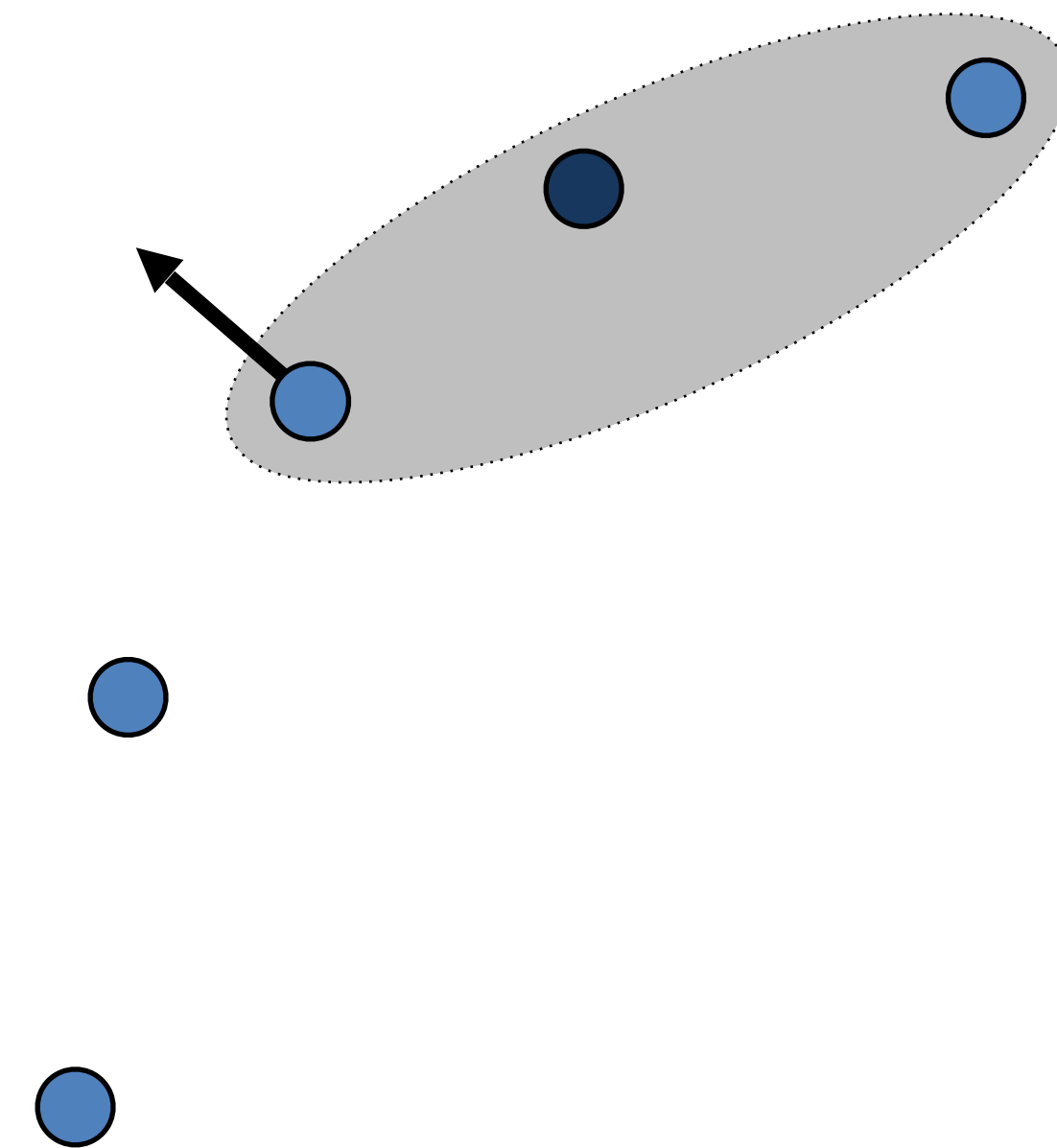
- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)





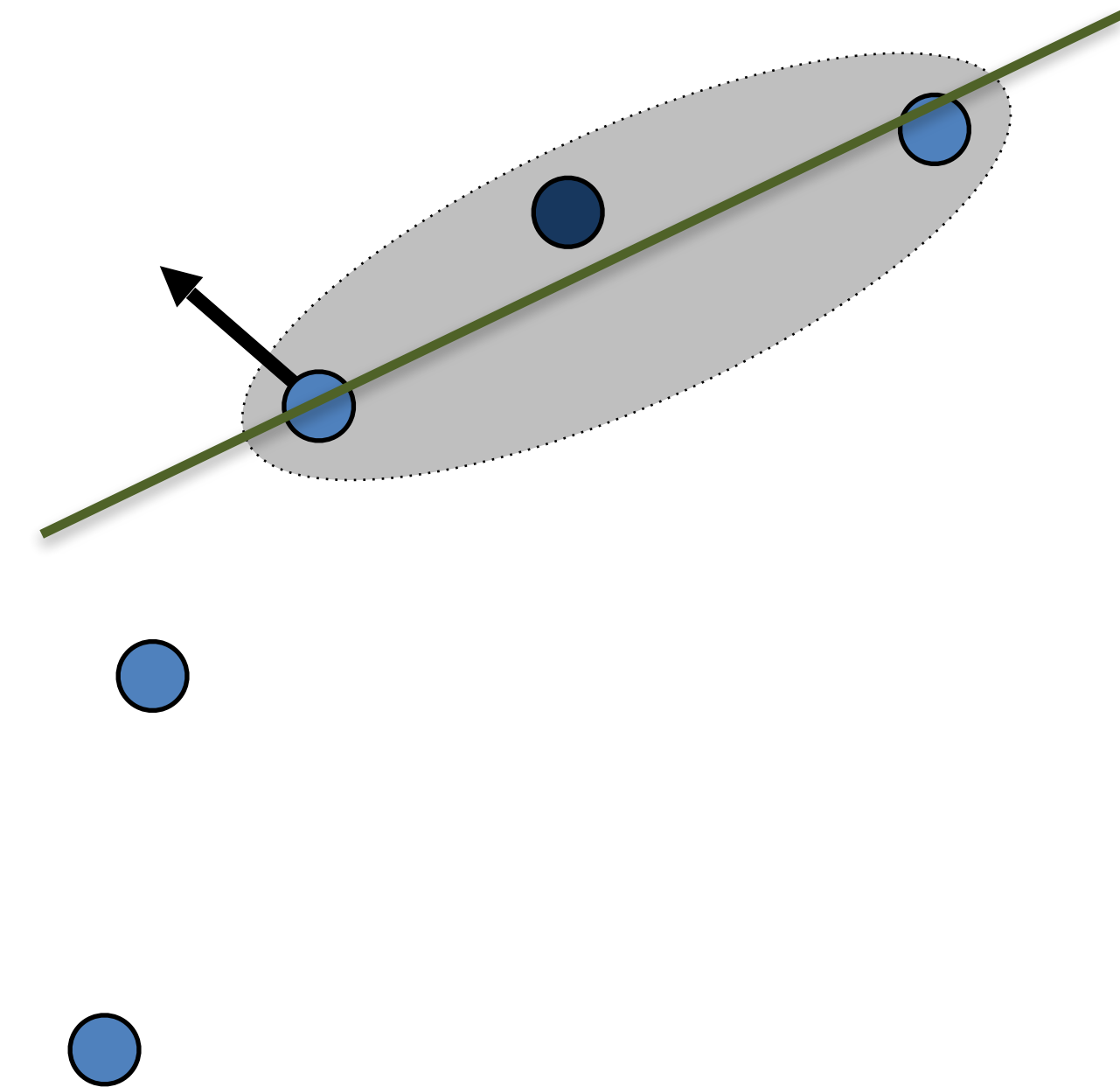
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



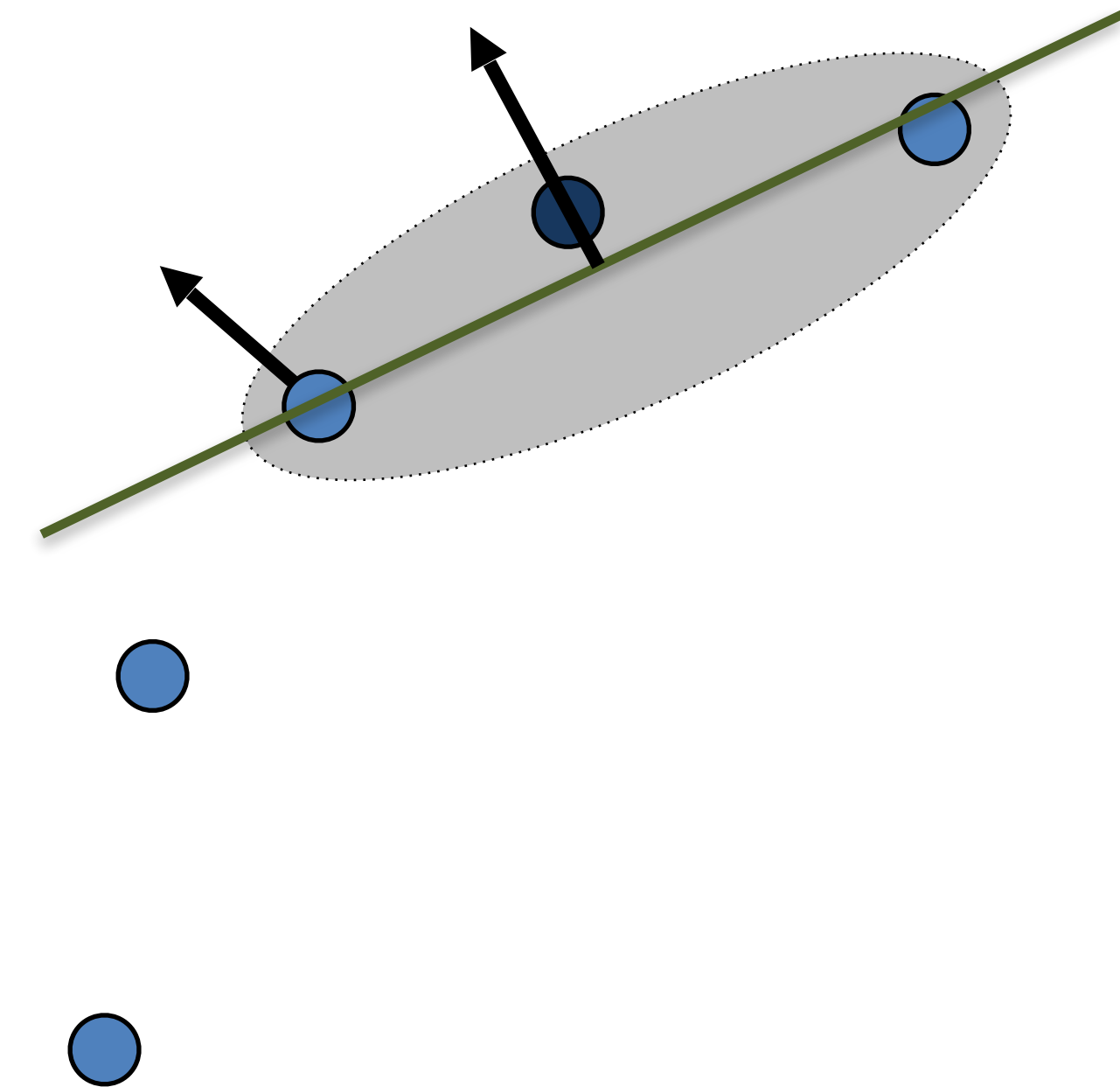
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



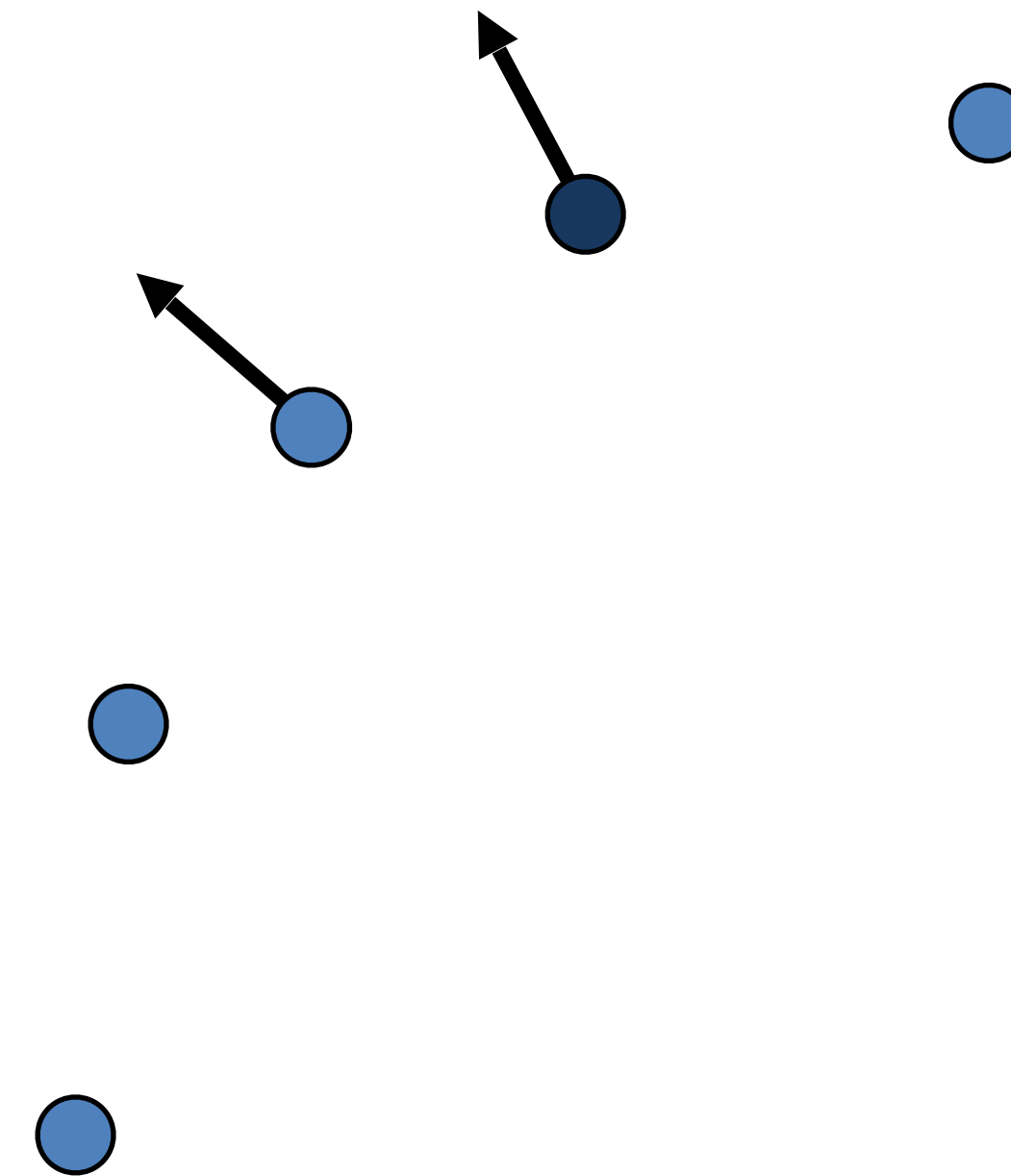
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



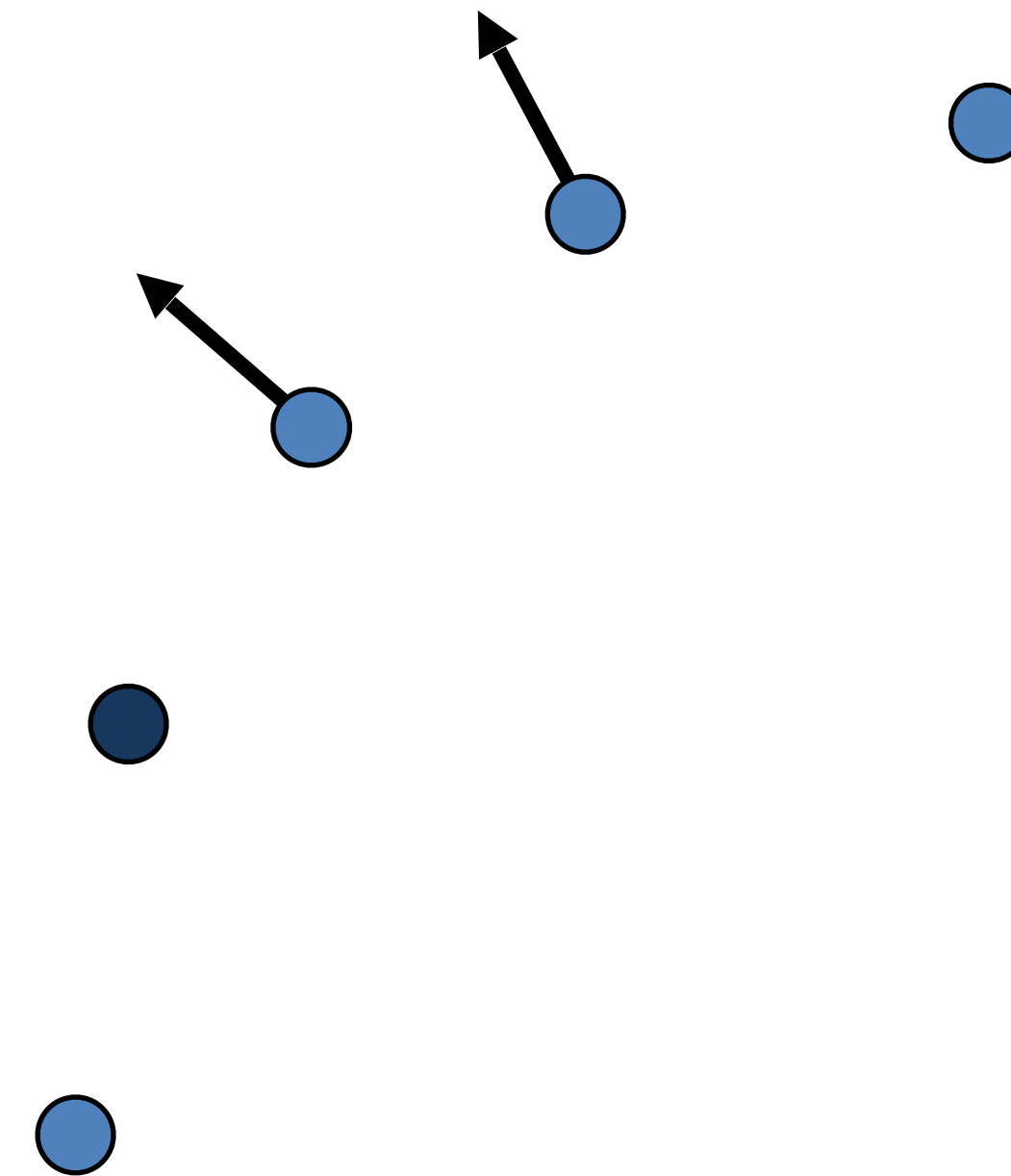
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



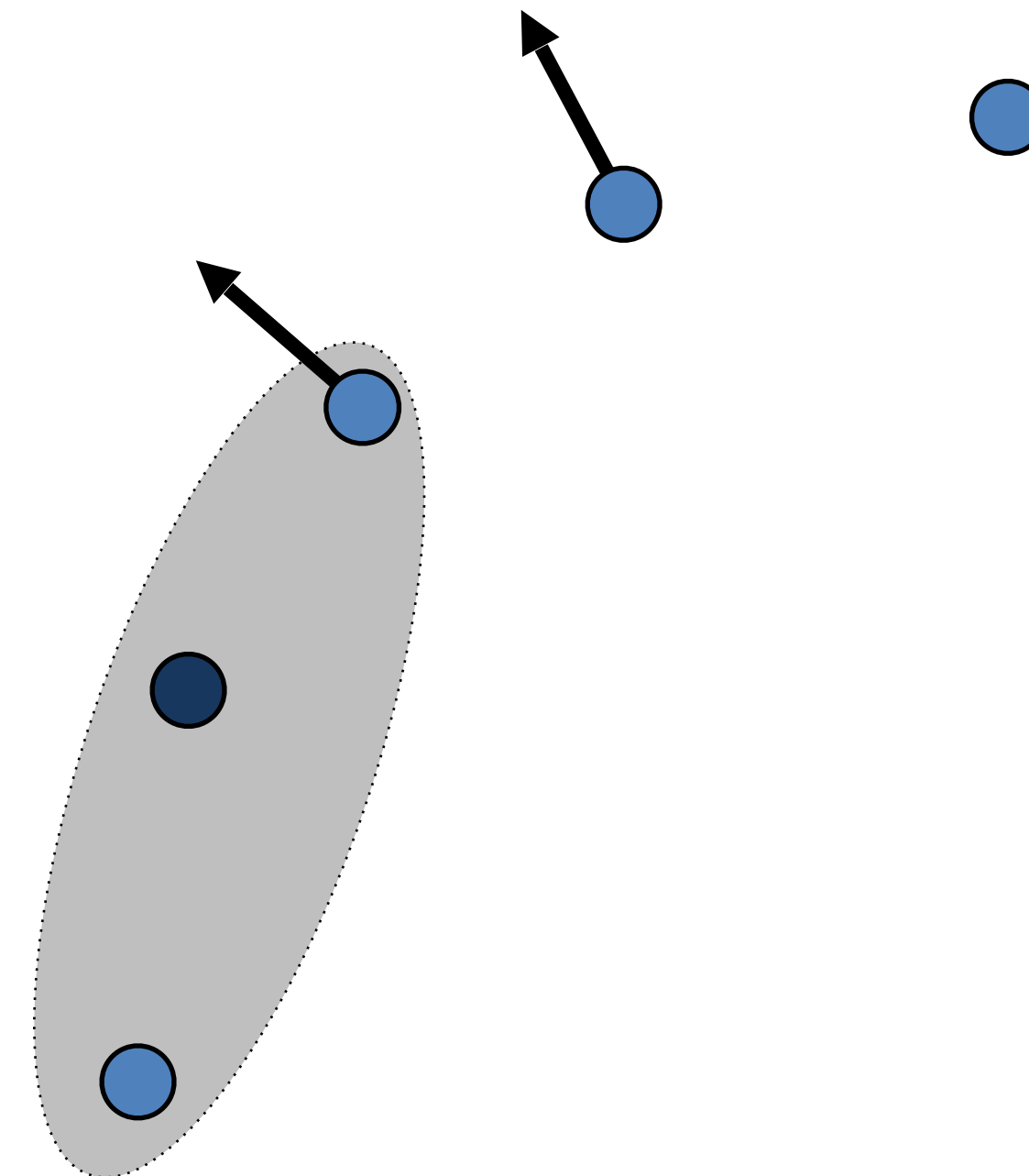
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



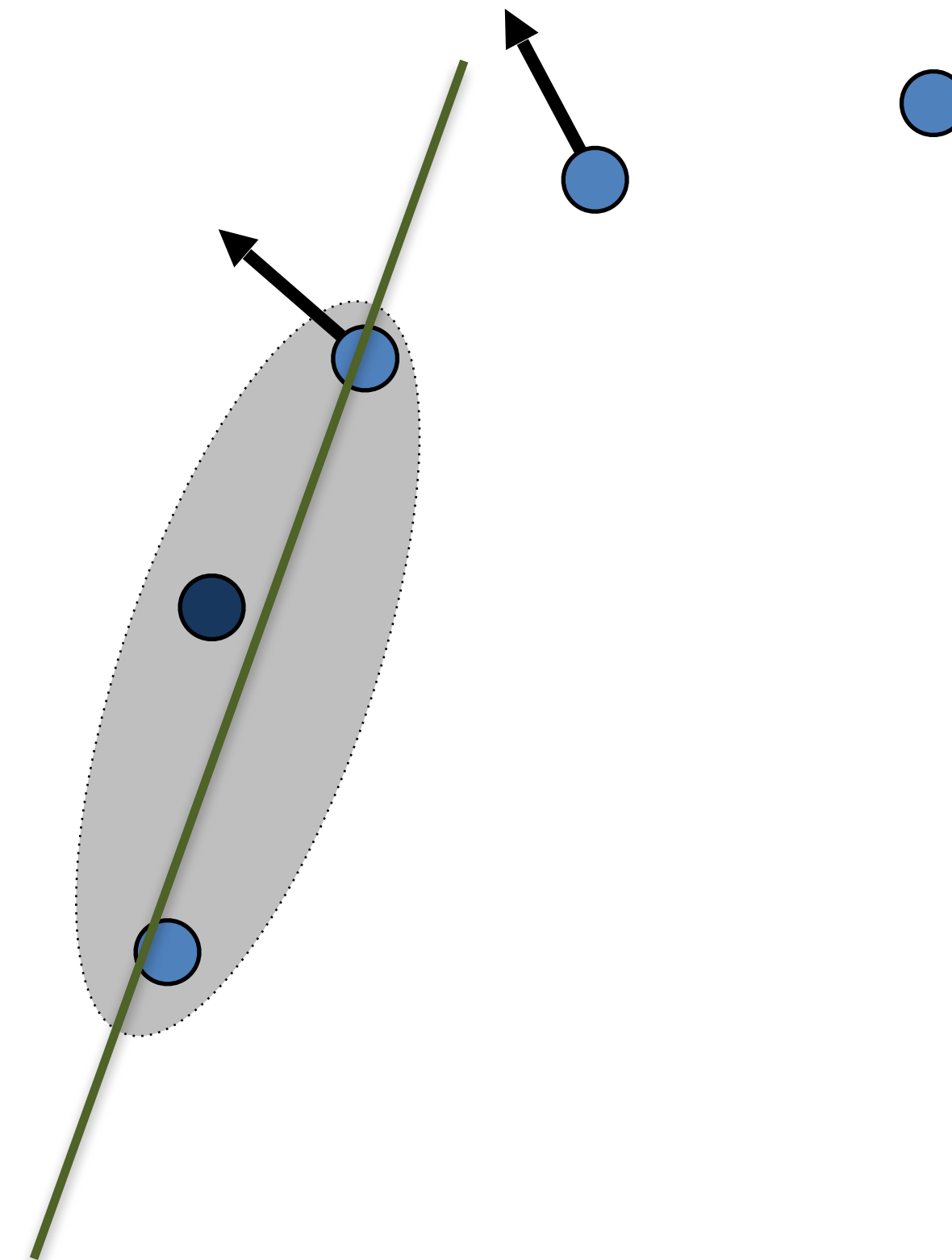
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



# Normal Estimation

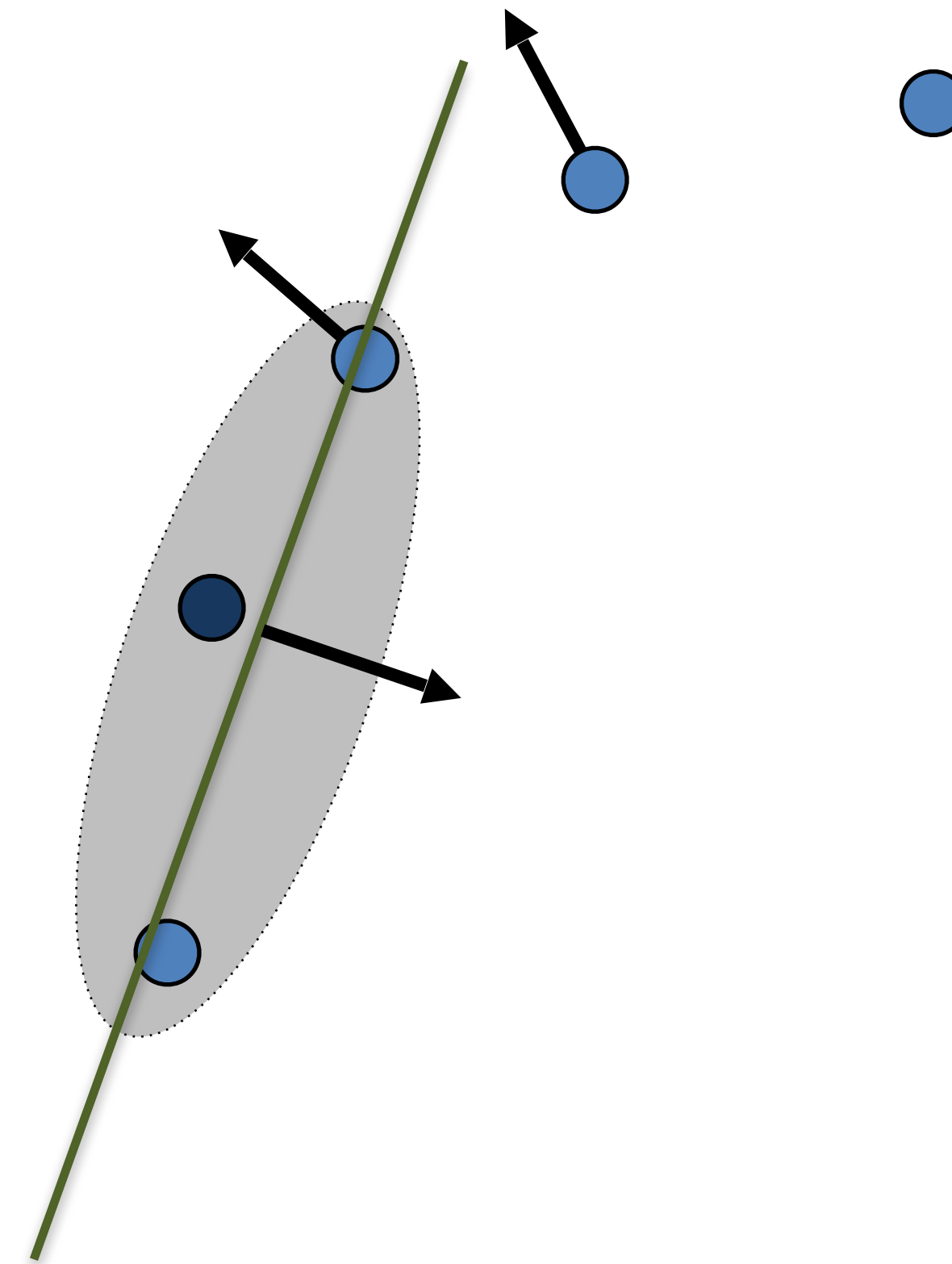
- Assign a normal vector  $\mathbf{n}$  at each point cloud point
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)





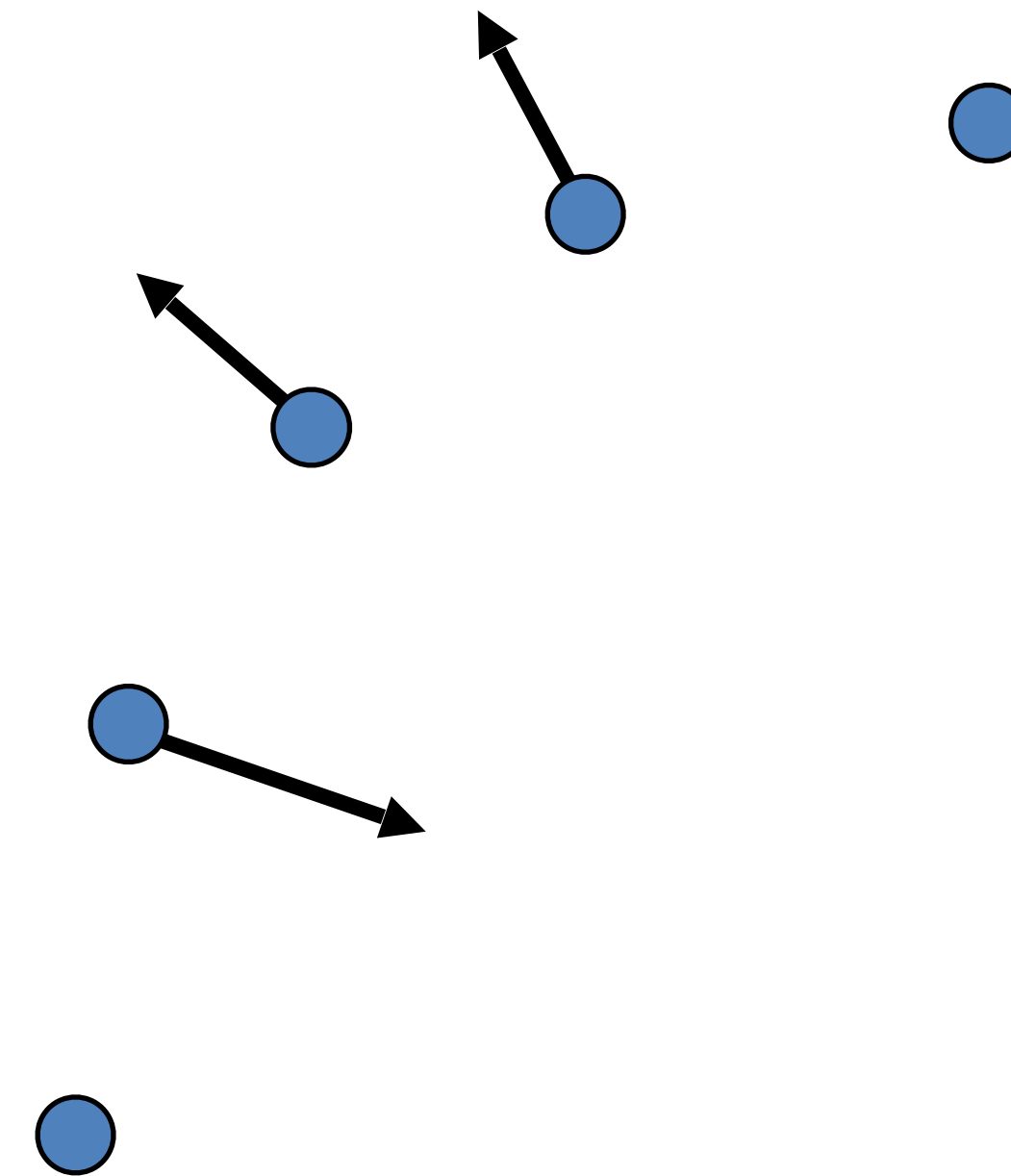
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



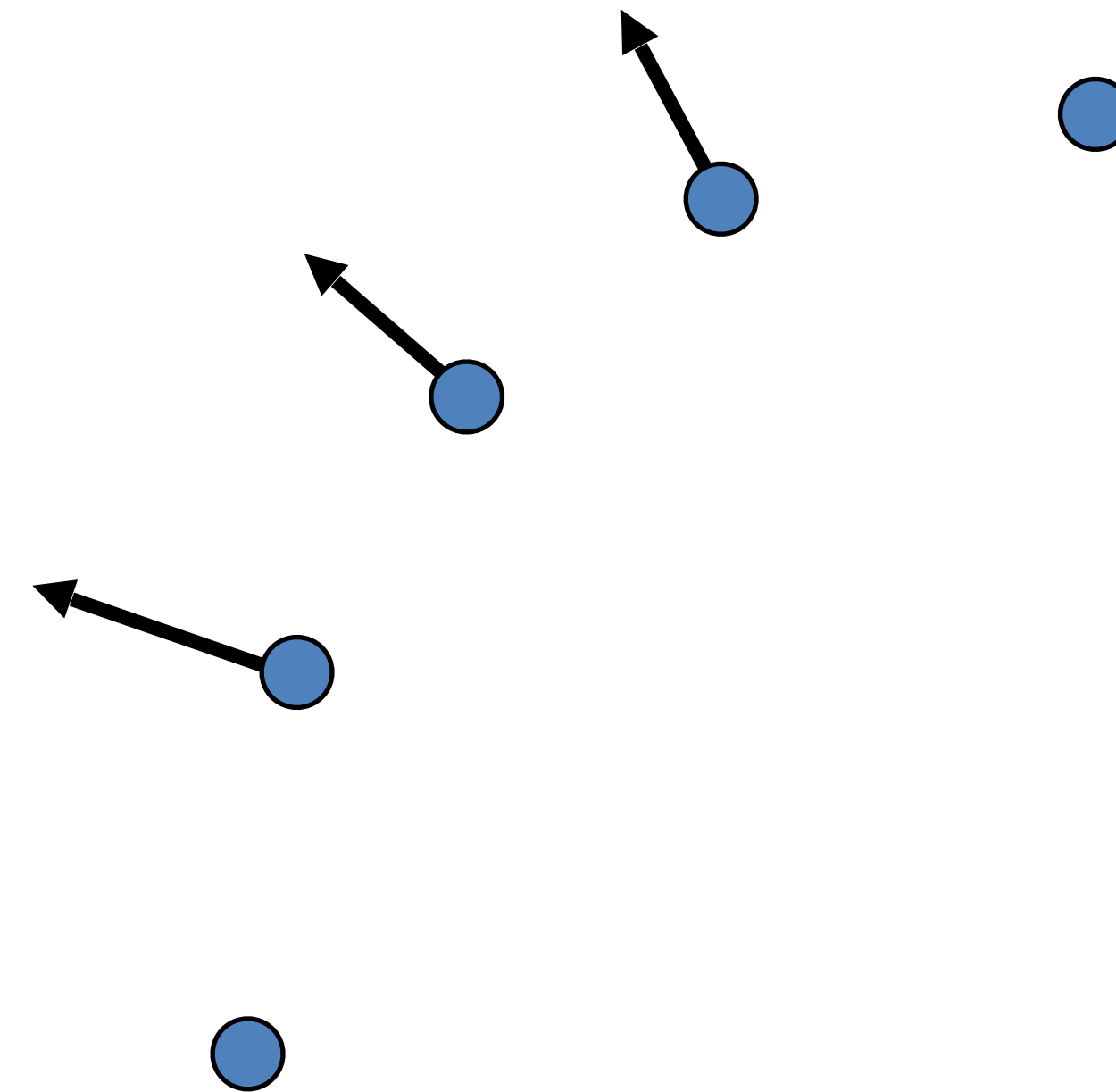
# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



# Normal Estimation

- Assign a normal vector  $\mathbf{n}$  at each point cloud point  $\mathbf{x}$ 
  - Estimate the direction by fitting a local plane
  - Find consistent global orientation by propagation (spanning tree)



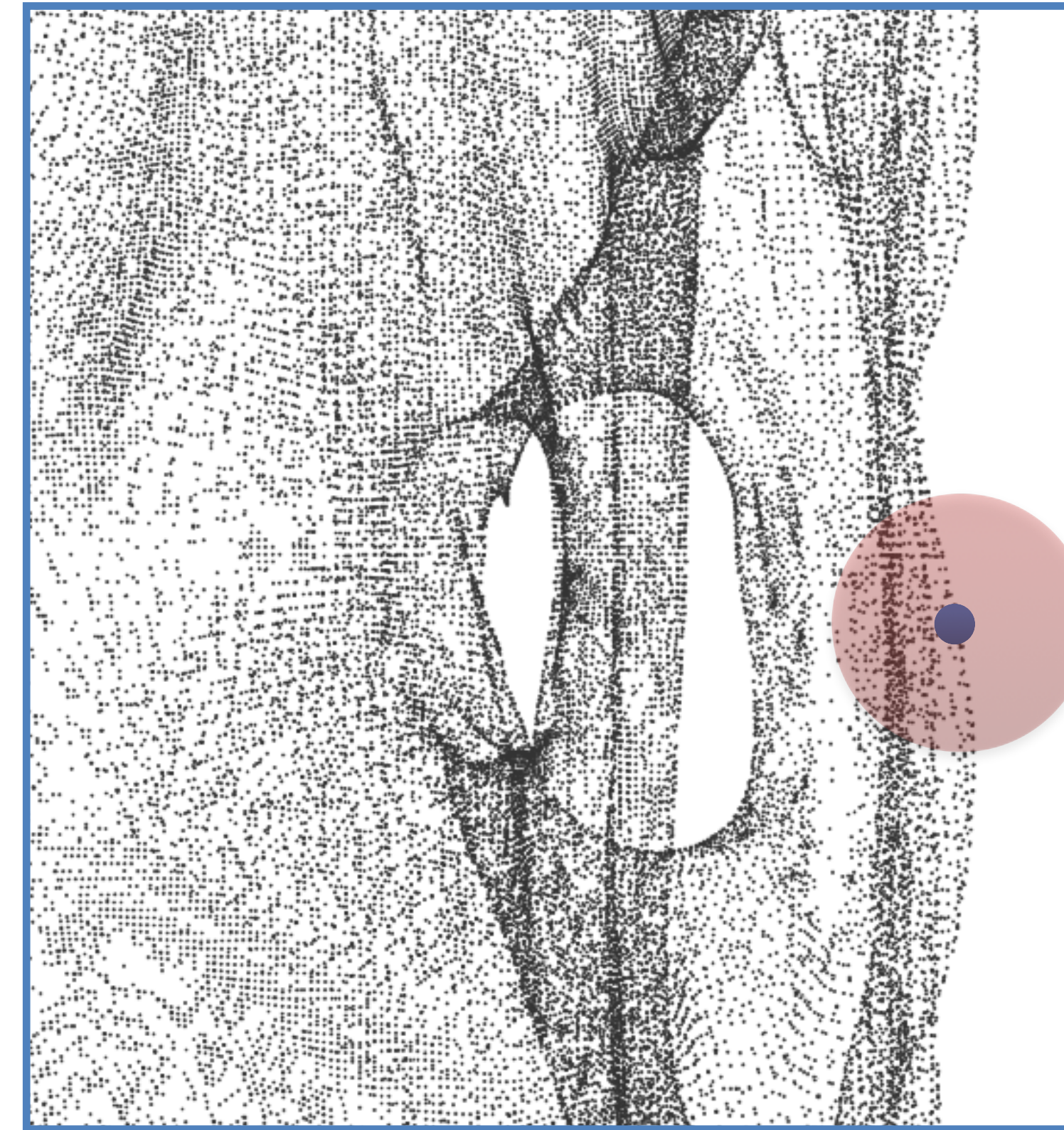
# Local Plane Fitting

- For each point  $\mathbf{x}$  in the cloud, pick  $k$  nearest neighbors or all points in  $r$ -ball:  $\{\mathbf{x}_i \mid \|\mathbf{x}_i - \mathbf{x}\| < r\}$

$$\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$$

- Find a plane  $\Pi$  that minimizes the sum of square distances:

$$\min \sum_{i=1}^n \text{dist}(\mathbf{x}_i, \Pi)^2$$





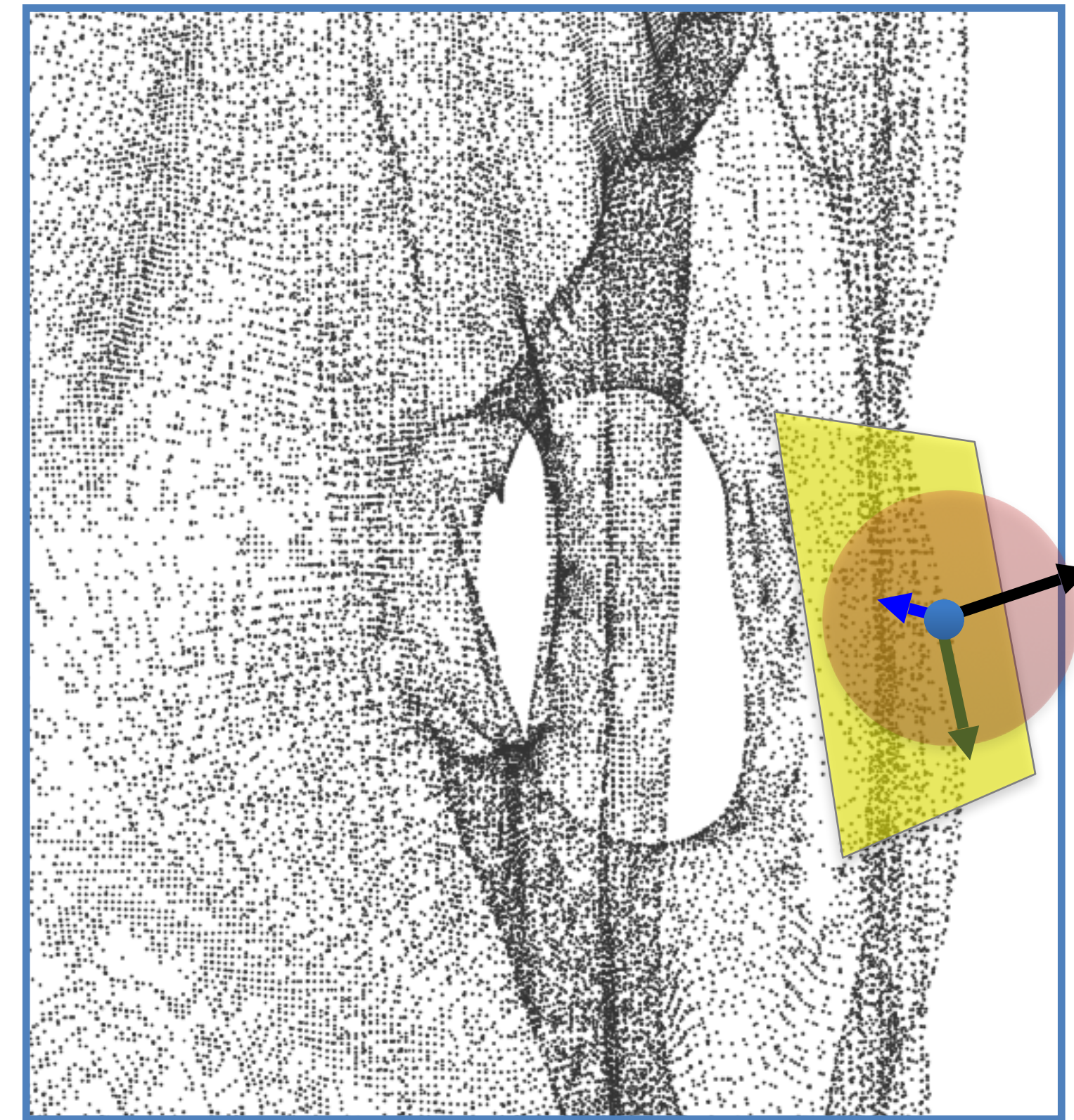
# Local Plane Fitting

- For each point  $\mathbf{x}$  in the cloud, pick  $k$  nearest neighbors or all points in  $r$ -ball:  $\{\mathbf{x}_i \mid \|\mathbf{x}_i - \mathbf{x}\| < r\}$

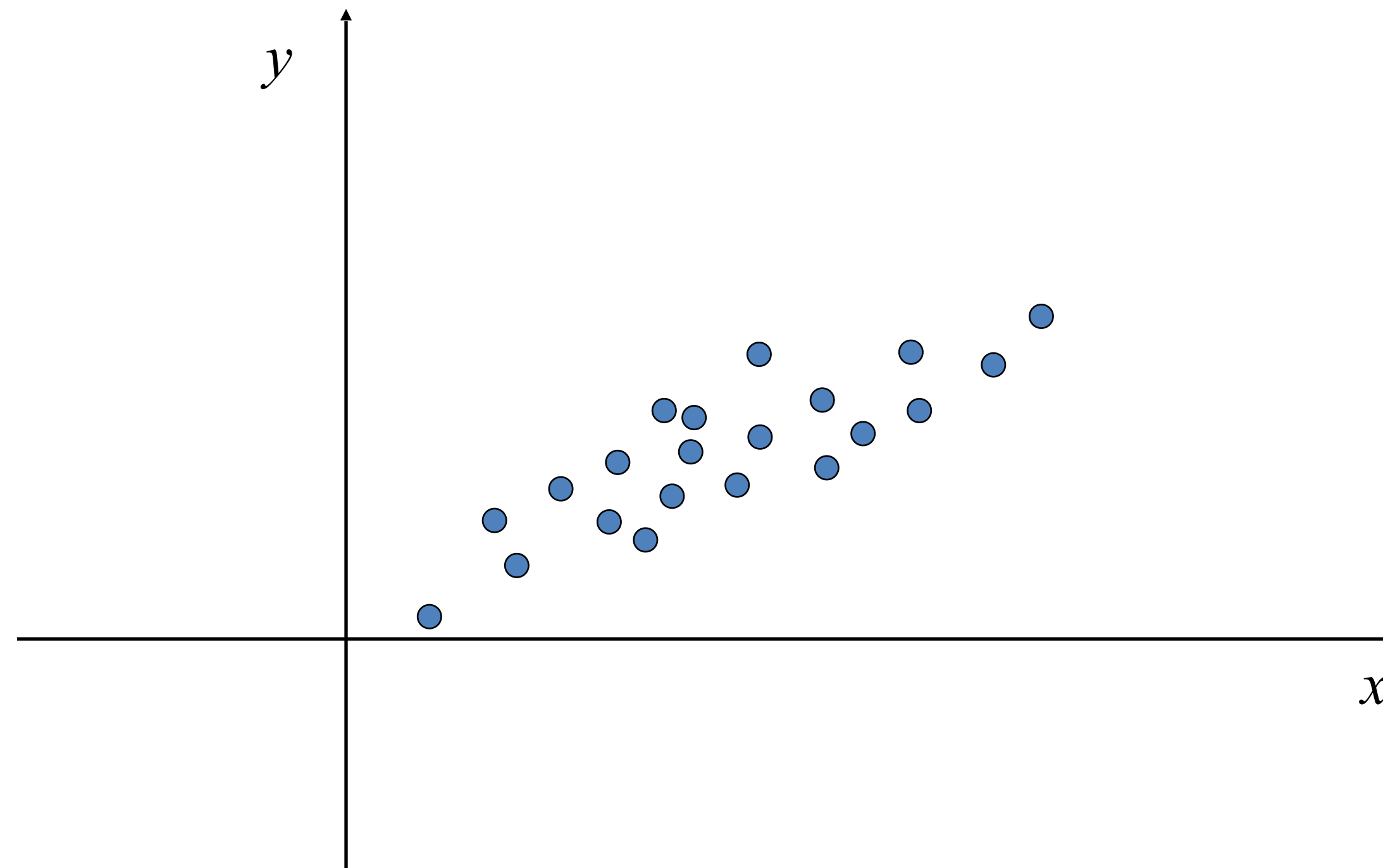
$$\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$$

- Find a plane  $\Pi$  that minimizes the sum of square distances:

$$\min \sum_{i=1}^n \text{dist}(\mathbf{x}_i, \Pi)^2$$

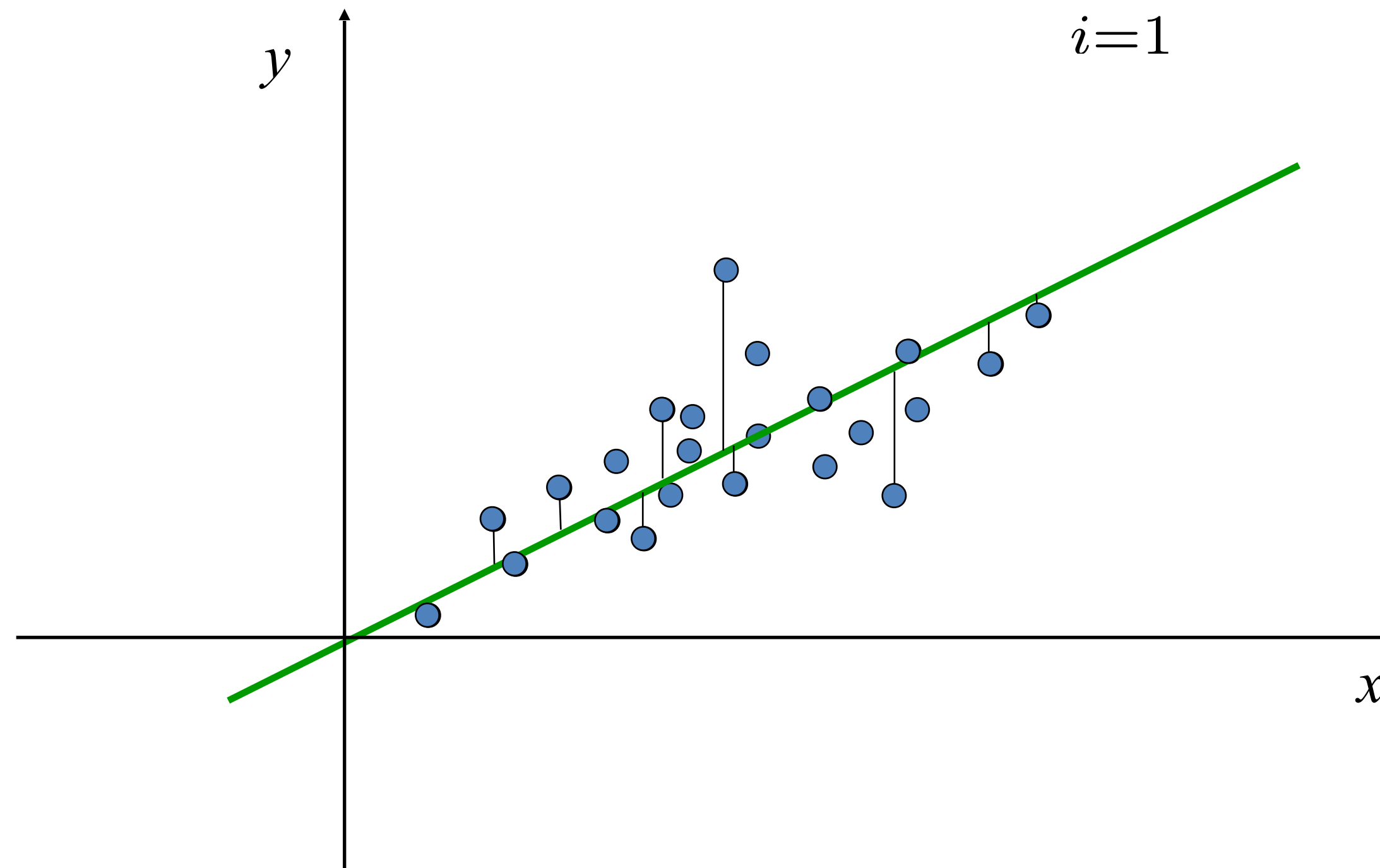


# Linear Least Squares?



# Linear Least Squares?

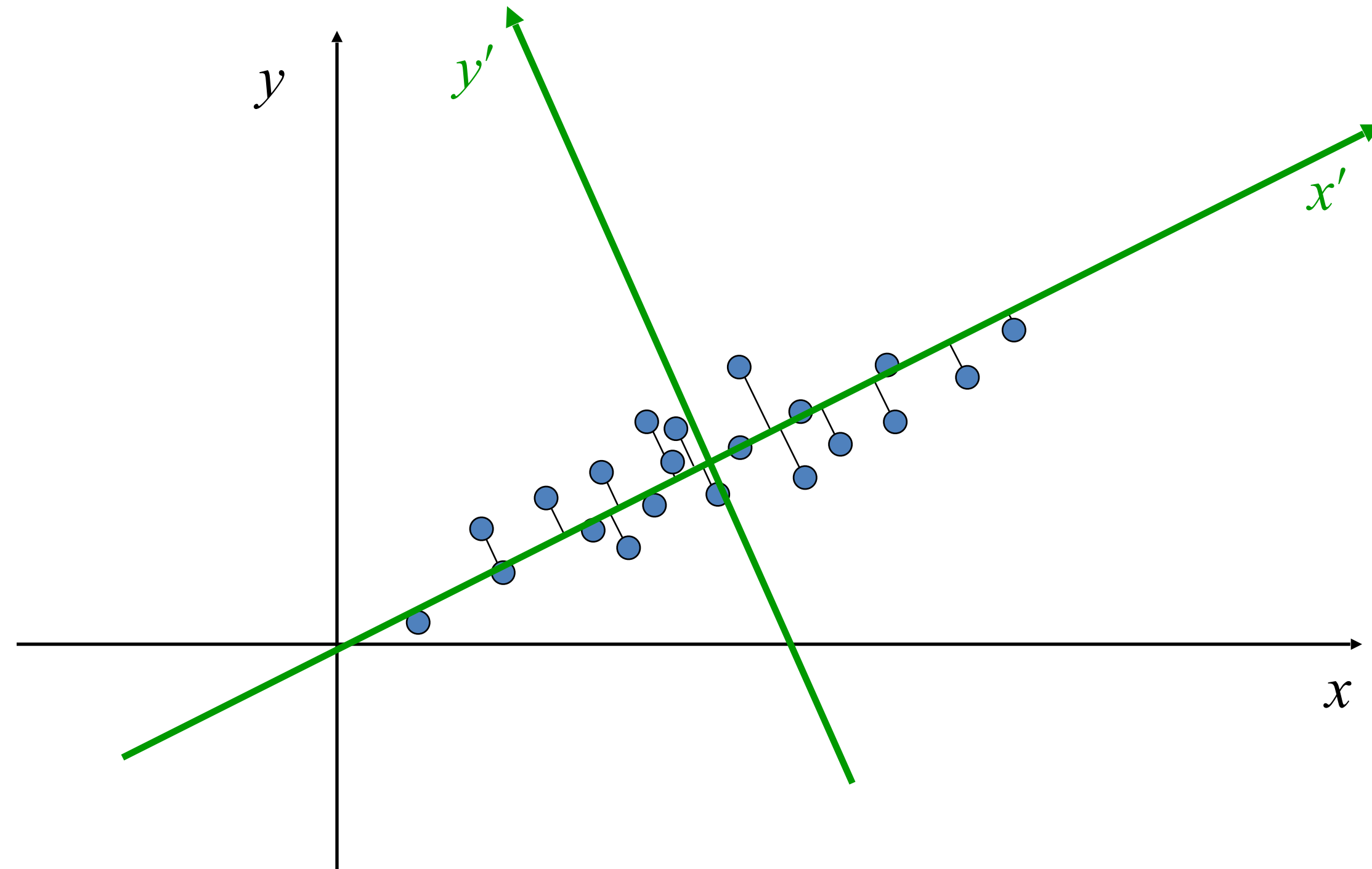
- Find a line  $y = ax + b$  s.t.  $\min \sum_{i=1}^n (y_i - (ax_i + b))^2$



- But we would like true orthogonal distances!



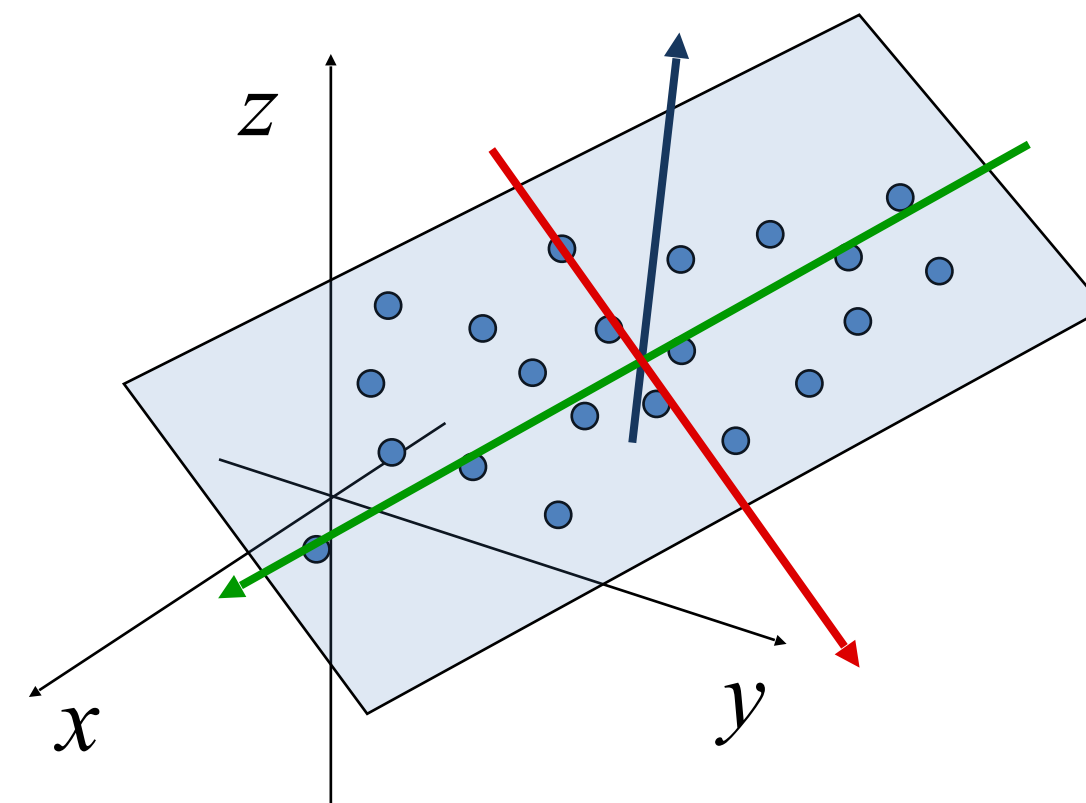
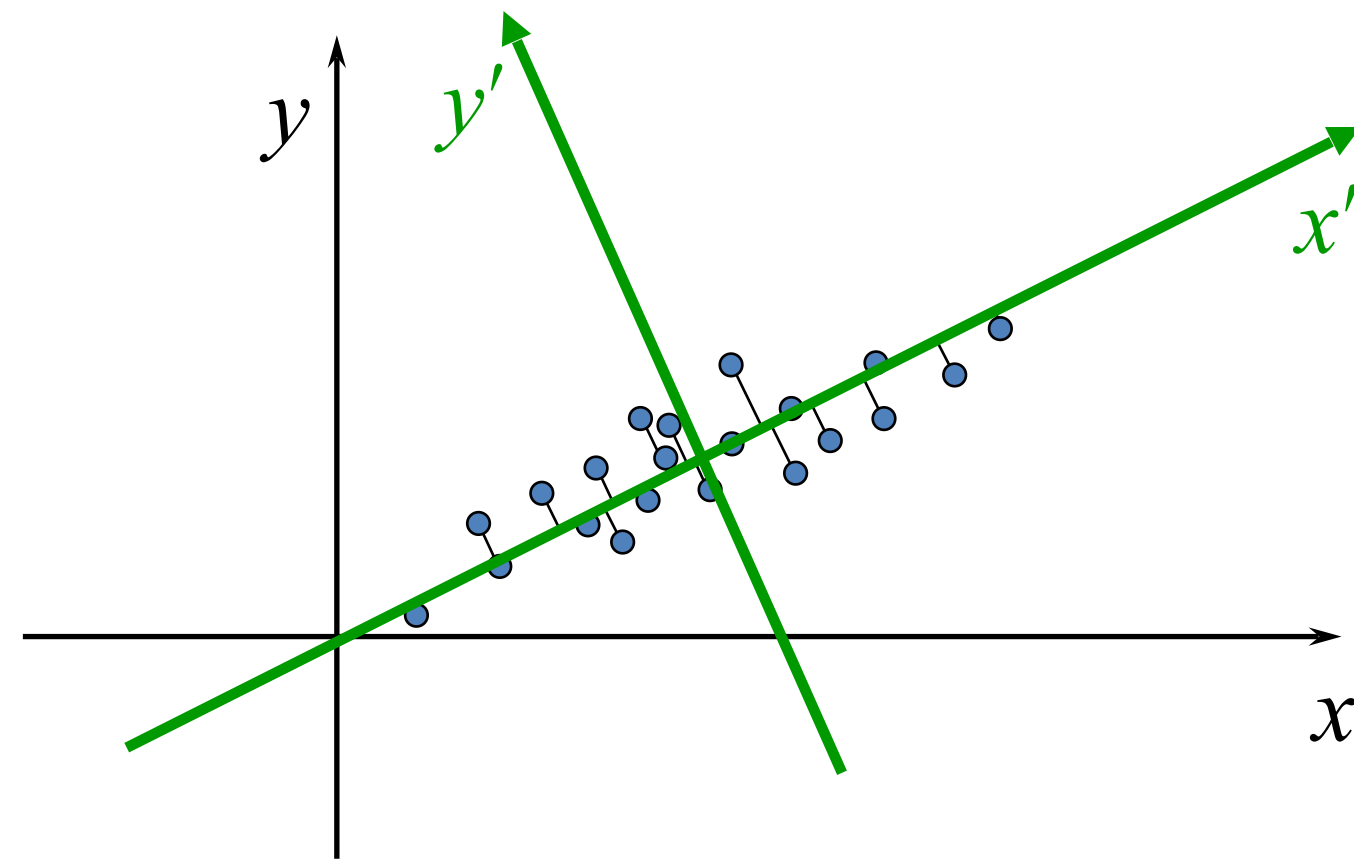
# Best Fit with SSD



SSD = sum of squared distances (or differences)

# Principle Component Analysis (PCA)

- PCA finds an orthogonal basis that best represents a given data set

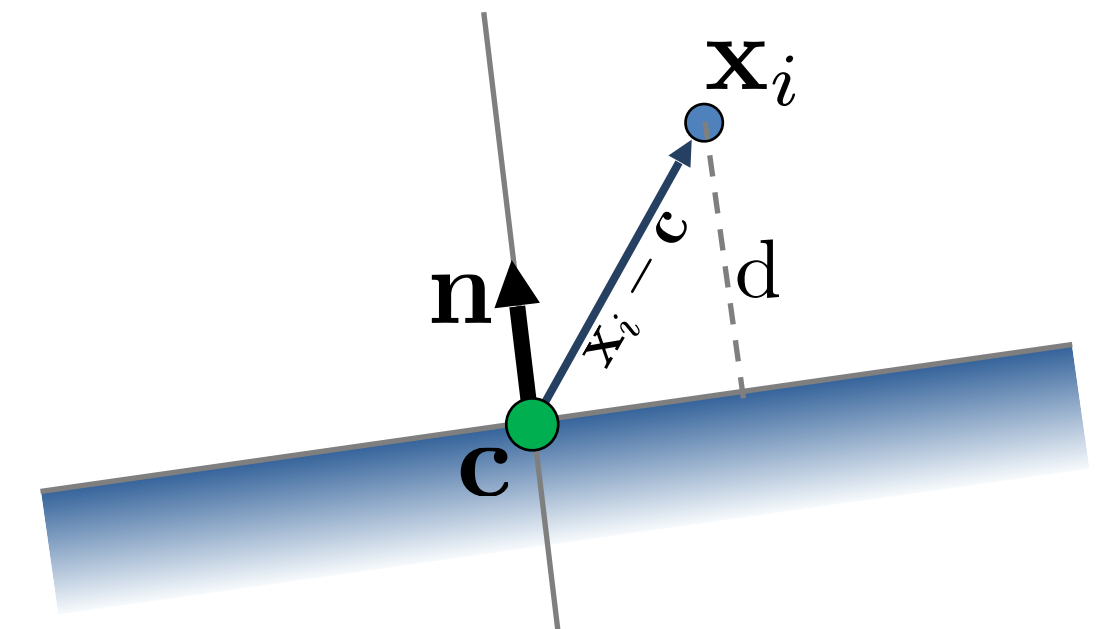
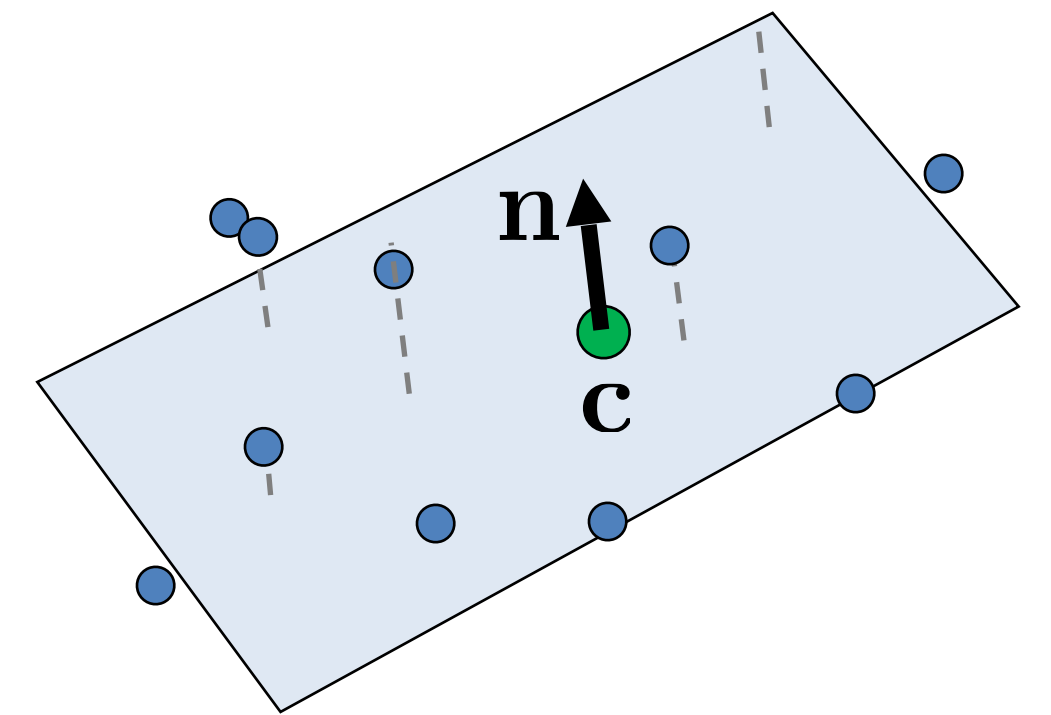


- PCA finds the best approximating line/plane/orientation... (in terms of  $\sum distances^2$ )

# Notations

- Input points:  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$
- Looking for a (hyper) plane passing through  $\mathbf{c}$  with normal  $\mathbf{n}$  s.t.

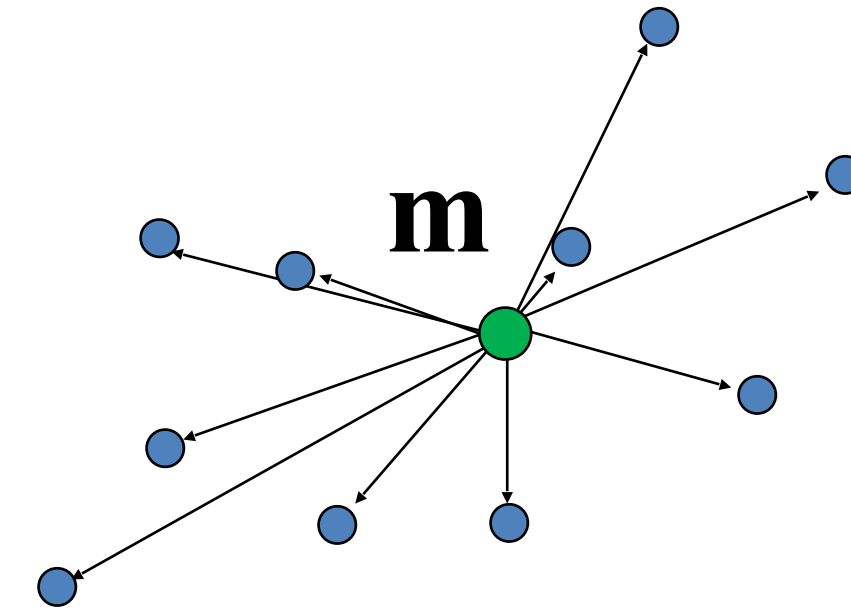
$$\min_{\mathbf{c}, \mathbf{n}, \|\mathbf{n}\|=1} \sum_{i=1}^n \left( (\mathbf{x}_i - \mathbf{c})^T \mathbf{n} \right)^2$$



# Notations

- Input points:  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$

- Centroid: 
$$\mathbf{m} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$



- Vectors from the centroid:

$$\mathbf{y}_i = \mathbf{x}_i - \mathbf{m}$$

# Centroid: 0-dim Approximation

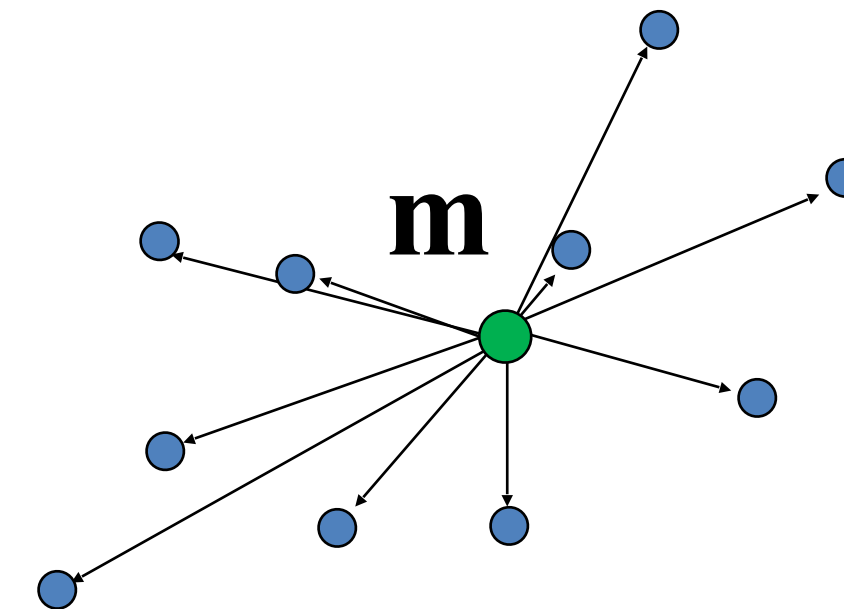
- It can be shown that:

$$\mathbf{m} = \operatorname{argmin}_{\mathbf{c}} \sum_{i=1}^n \left( (\mathbf{x}_i - \mathbf{c})^T \mathbf{n} \right)^2$$

$$\mathbf{m} = \operatorname{argmin}_{\mathbf{c}} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{c}\|^2$$

- $\mathbf{m}$  minimizes SSD
- $\mathbf{m}$  will be the origin of the (hyper)-plane

- Our problem becomes:  $\min_{\|\mathbf{n}\|=1} \sum_{i=1}^n \left( \mathbf{y}_i^T \mathbf{n} \right)^2$



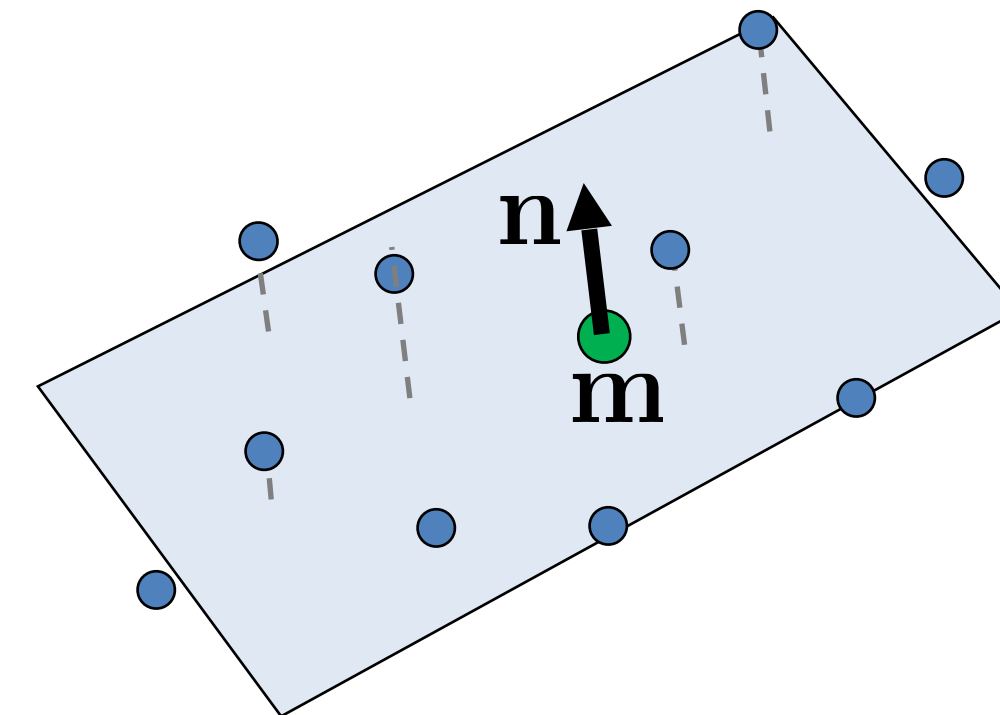
$$\mathbf{m} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$

# Hyperplane Normal

- Minimize!

$$\begin{aligned}\min_{\mathbf{n}^T \mathbf{n} = 1} \sum_{i=1}^n (\mathbf{y}_i^T \mathbf{n})^2 &= \min_{\mathbf{n}^T \mathbf{n} = 1} \sum_{i=1}^n \mathbf{n}^T \mathbf{y}_i \mathbf{y}_i^T \mathbf{n} = \\ \min_{\mathbf{n}^T \mathbf{n} = 1} \mathbf{n}^T \left( \sum_{i=1}^n \mathbf{y}_i \mathbf{y}_i^T \right) \mathbf{n} &= \min_{\mathbf{n}^T \mathbf{n} = 1} \mathbf{n}^T (\mathbf{Y} \mathbf{Y}^T) \mathbf{n}\end{aligned}$$

$$\mathbf{Y} = \begin{pmatrix} | & | & \dots & | \\ \mathbf{y}_1 & \mathbf{y}_2 & & \mathbf{y}_n \\ | & | & & | \end{pmatrix}$$



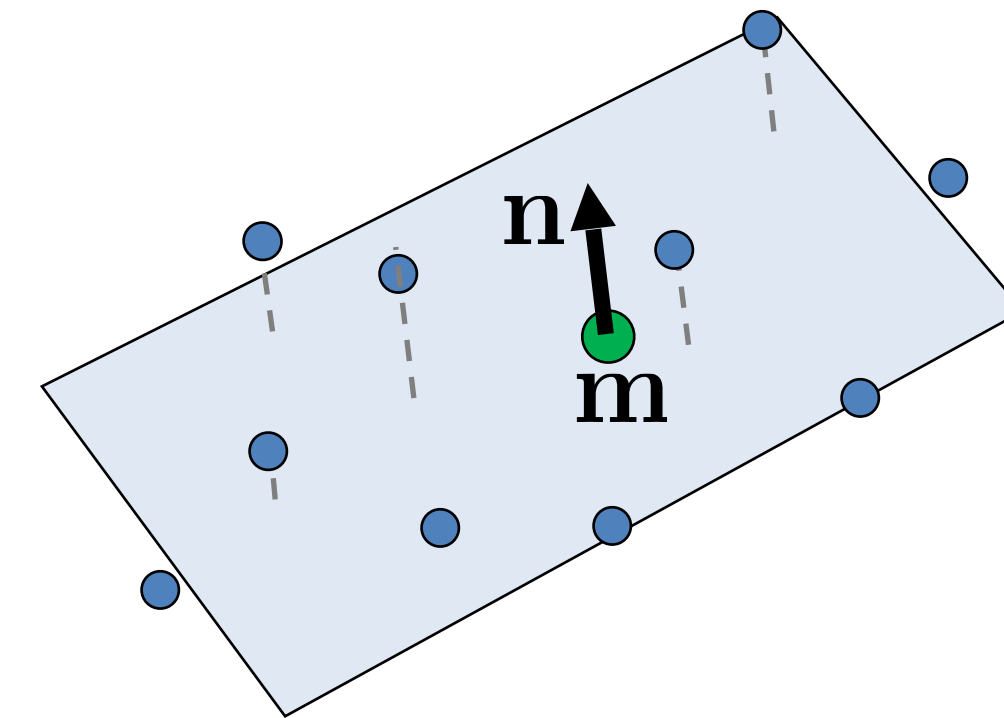
# Hyperplane Normal

- Minimize!

$$\begin{aligned}\min_{\mathbf{n}^T \mathbf{n} = 1} \sum_{i=1}^n (\mathbf{y}_i^T \mathbf{n})^2 &= \min_{\mathbf{n}^T \mathbf{n} = 1} \sum_{i=1}^n \mathbf{n}^T \mathbf{y}_i \mathbf{y}_i^T \mathbf{n} = \\ \min_{\mathbf{n}^T \mathbf{n} = 1} \mathbf{n}^T \left( \sum_{i=1}^n \mathbf{y}_i \mathbf{y}_i^T \right) \mathbf{n} &= \min_{\mathbf{n}^T \mathbf{n} = 1} \mathbf{n}^T (\mathbf{Y} \mathbf{Y}^T) \mathbf{n}\end{aligned}$$

$$\mathbf{Y} = \begin{pmatrix} | & | & \dots & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_n \\ | & | & \dots & | \end{pmatrix}$$

$$\begin{aligned}f(\mathbf{n}) &= \mathbf{n}^T \mathbf{S} \mathbf{n} \quad (\mathbf{S} = \mathbf{Y} \mathbf{Y}^T) \\ \min f(\mathbf{n}) \quad s.t. \quad \mathbf{n}^T \mathbf{n} &= 1\end{aligned}$$





# Hyperplane Normal

- Constrained minimization – Lagrange multipliers

$$f(\mathbf{n}) = \mathbf{n}^T \mathbf{S} \mathbf{n} \quad (\mathbf{S} = \mathbf{Y} \mathbf{Y}^T)$$
$$\min f(\mathbf{n}) \quad s.t. \quad \mathbf{n}^T \mathbf{n} = 1$$

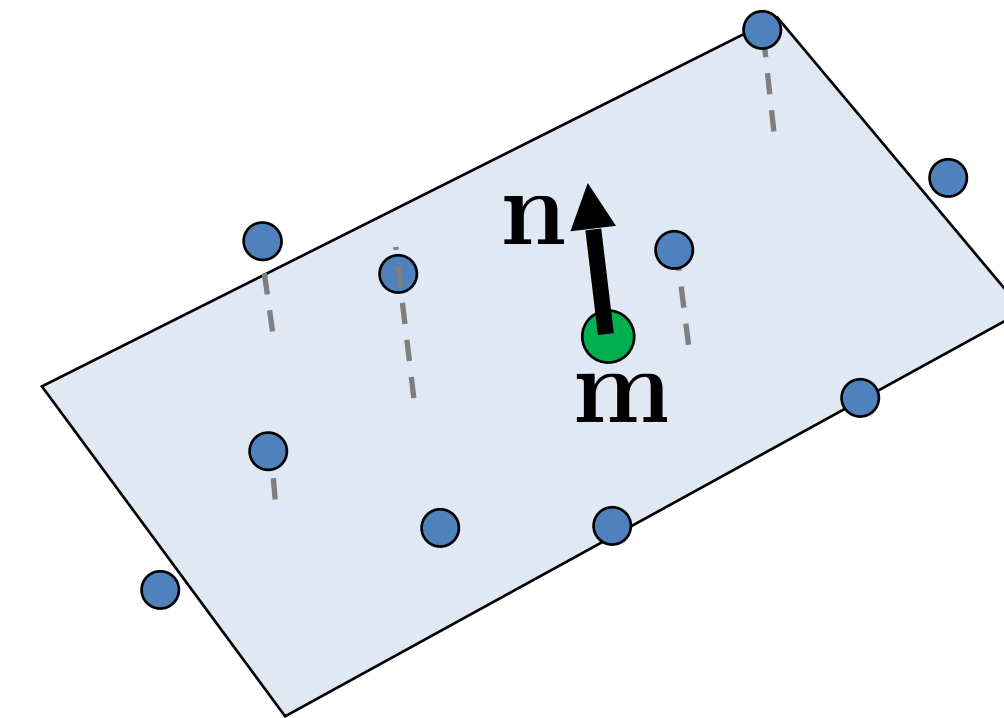
$$\mathcal{L}(\mathbf{n}, \lambda) = f(\mathbf{n}) - \lambda(\mathbf{n}^T \mathbf{n} - 1)$$

$$\nabla \mathcal{L} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}} = \frac{\partial}{\partial \mathbf{n}} f(\mathbf{n}) - \lambda \frac{\partial}{\partial \mathbf{n}} (\mathbf{n}^T \mathbf{n} - 1)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \mathbf{n}^T \mathbf{n} - 1$$

$$\frac{\partial}{\partial \mathbf{n}} f(\mathbf{n}) - \lambda \frac{\partial}{\partial \mathbf{n}} (\mathbf{n}^T \mathbf{n} - 1) = (\mathbf{S} + \mathbf{S}^T) \mathbf{n} - \lambda(\mathbf{I} + \mathbf{I}^T) \mathbf{n} = 2\mathbf{S} \mathbf{n} - 2\lambda \mathbf{n}$$



Derivatives: see Matrix Cookbook  
<https://archive.org/details/imm3274>

# Hyperplane Normal

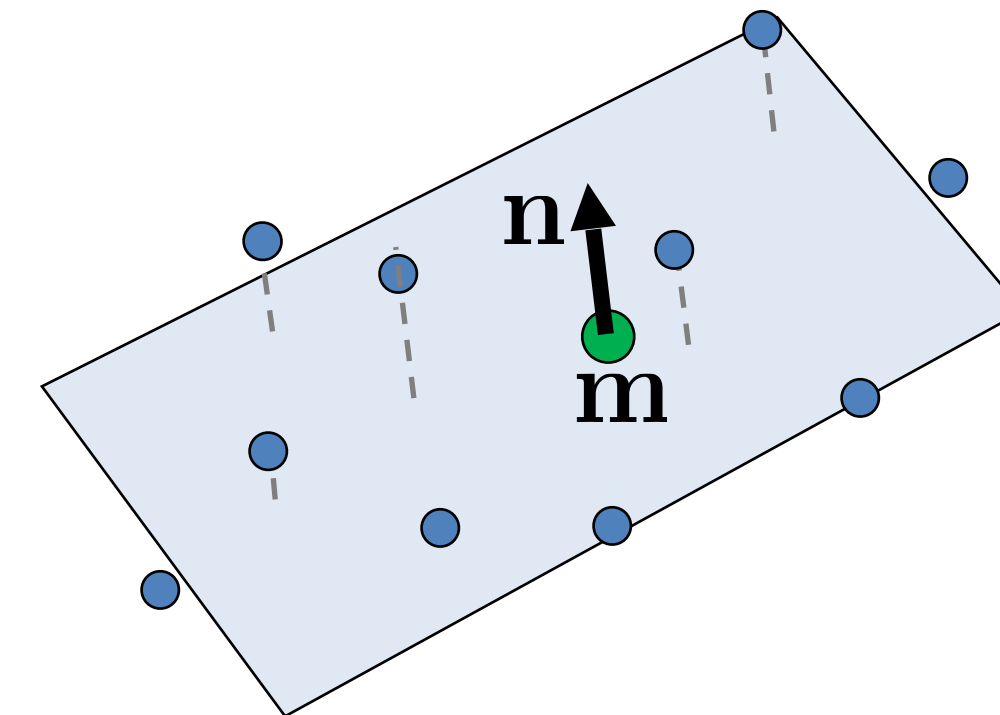
- Constrained minimization – Lagrange multipliers

$$f(\mathbf{n}) = \mathbf{n}^T \mathbf{S} \mathbf{n} \quad (\mathbf{S} = \mathbf{Y} \mathbf{Y}^T)$$
$$\min f(\mathbf{n}) \quad s.t. \quad \mathbf{n}^T \mathbf{n} = 1$$

$$\mathcal{L}(\mathbf{n}, \lambda) = f(\mathbf{n}) - \lambda(\mathbf{n}^T \mathbf{n} - 1)$$

$$\nabla \mathcal{L} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}} = 0 \iff \mathbf{S} \mathbf{n} = \lambda \mathbf{n}$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \iff \mathbf{n}^T \mathbf{n} = 1$$



# Hyperplane Normal

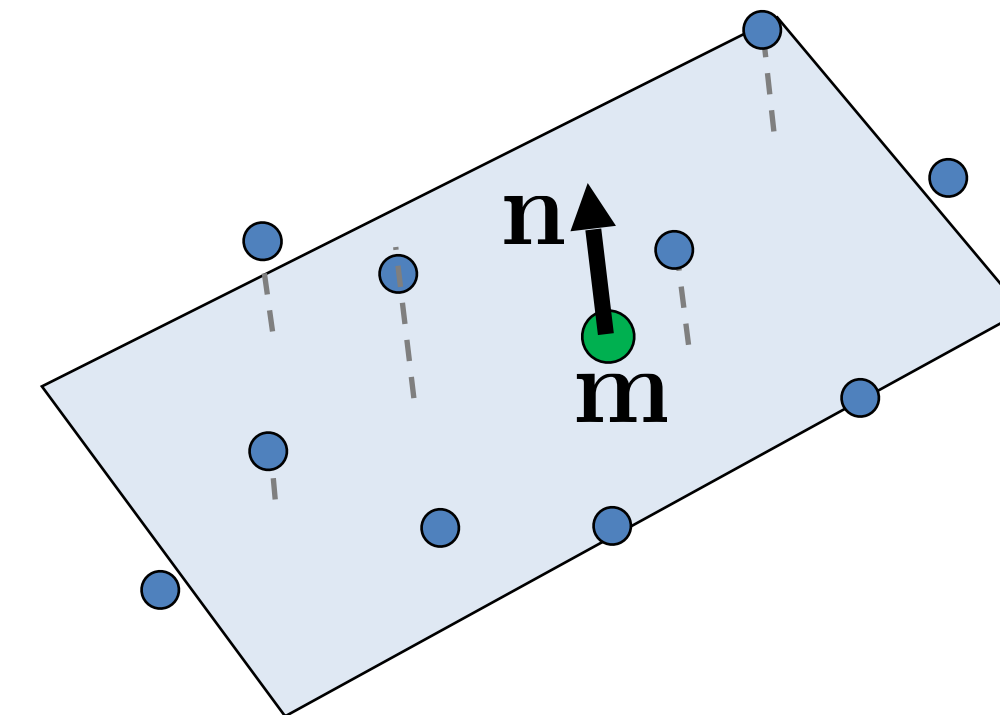
- Constrained minimization – Lagrange multipliers

$$f(\mathbf{n}) = \mathbf{n}^T \mathbf{S} \mathbf{n} \quad (\mathbf{S} = \mathbf{Y} \mathbf{Y}^T)$$
$$\min f(\mathbf{n}) \quad s.t. \quad \mathbf{n}^T \mathbf{n} = 1$$

$$\mathcal{L}(\mathbf{n}, \lambda) = f(\mathbf{n}) - \lambda(\mathbf{n}^T \mathbf{n} - 1)$$

$$\nabla \mathcal{L} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}} = 0 \iff \mathbf{S} \mathbf{n} = \lambda \mathbf{n}$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \iff \mathbf{n}^T \mathbf{n} = 1$$



What can be said about  $\mathbf{n}$  ??

# Hyperplane Normal

- Constrained minimization – Lagrange multipliers

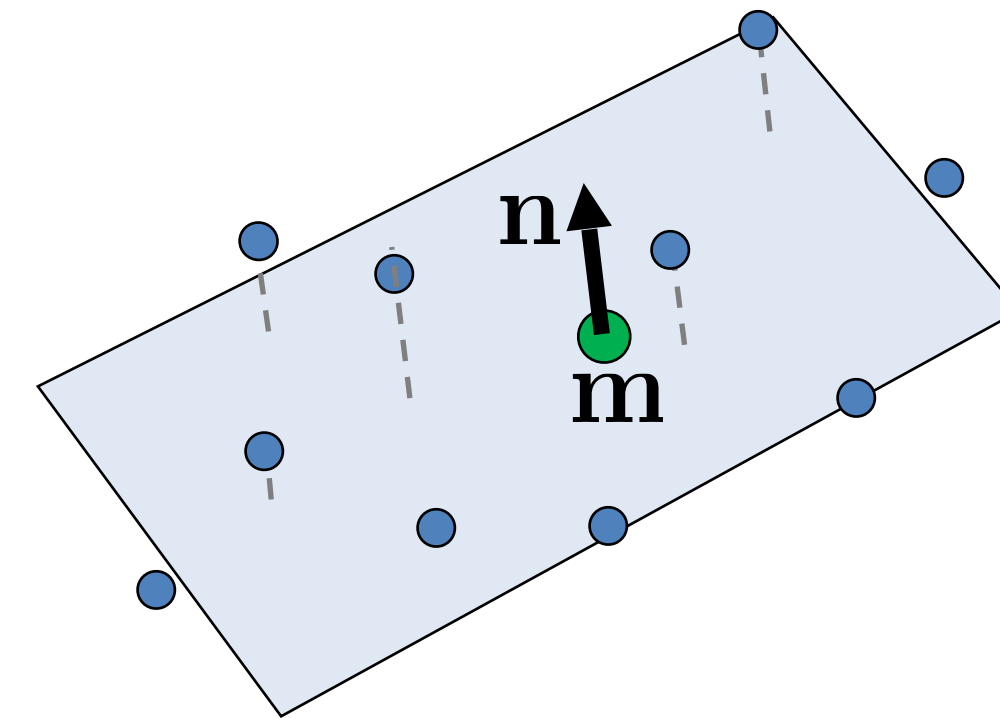
$$f(\mathbf{n}) = \mathbf{n}^T \mathbf{S} \mathbf{n} \quad (\mathbf{S} = \mathbf{Y} \mathbf{Y}^T)$$
$$\min f(\mathbf{n}) \quad s.t. \quad \mathbf{n}^T \mathbf{n} = 1$$

$$\mathcal{L}(\mathbf{n}, \lambda) = f(\mathbf{n}) - \lambda(\mathbf{n}^T \mathbf{n} - 1)$$

$$\nabla \mathcal{L} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}} = 0 \iff \mathbf{S} \mathbf{n} = \lambda \mathbf{n}$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \iff \mathbf{n}^T \mathbf{n} = 1$$

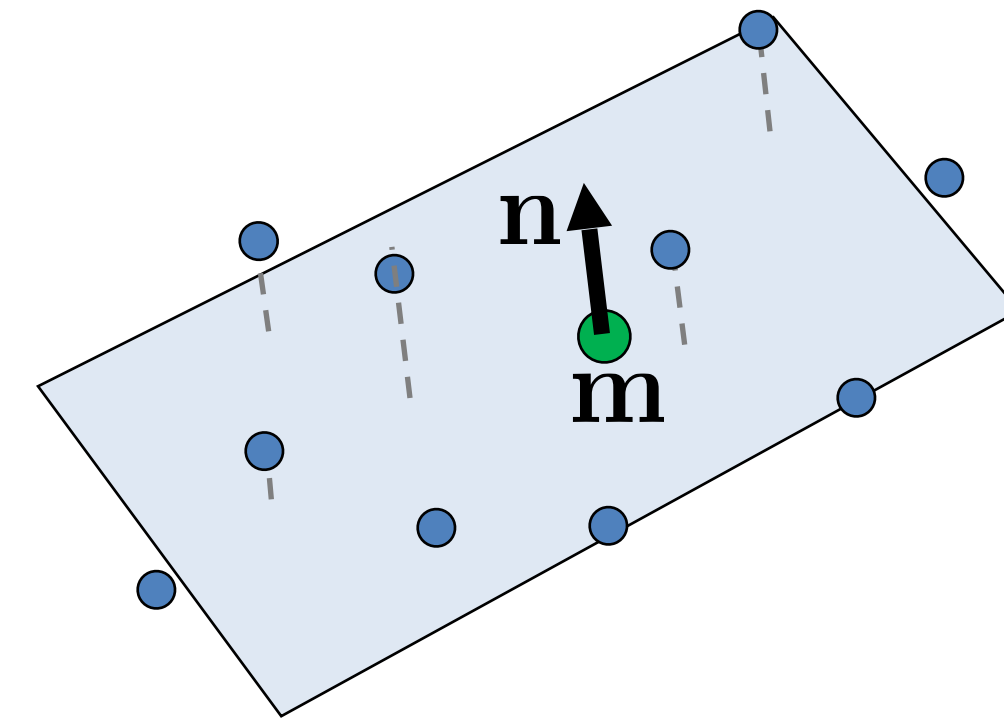
$\mathbf{n}$  is the eigenvector of  $\mathbf{S}$   
with the smallest  
eigenvalue



# Summary – Best Fitting Plane Recipe

- Input:  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$
- Compute centroid = plane origin  $\mathbf{m} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$
- Compute scatter matrix  $\mathbf{S} = \mathbf{Y}\mathbf{Y}^T$   
 $\mathbf{Y} = (\mathbf{y}_1 \ \mathbf{y}_2 \ \dots \ \mathbf{y}_n)$   
 $\mathbf{y}_i = \mathbf{x}_i - \mathbf{m}$
- The plane normal  $\mathbf{n}$  is the eigenvector of  $\mathbf{S}$  with the smallest eigenvalue

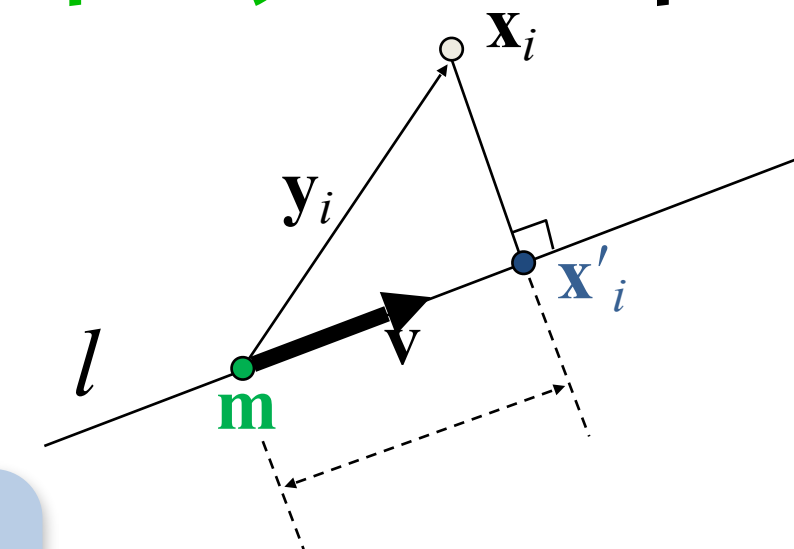
$$\mathbf{S} = \mathbf{V} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix} \mathbf{V}^T$$



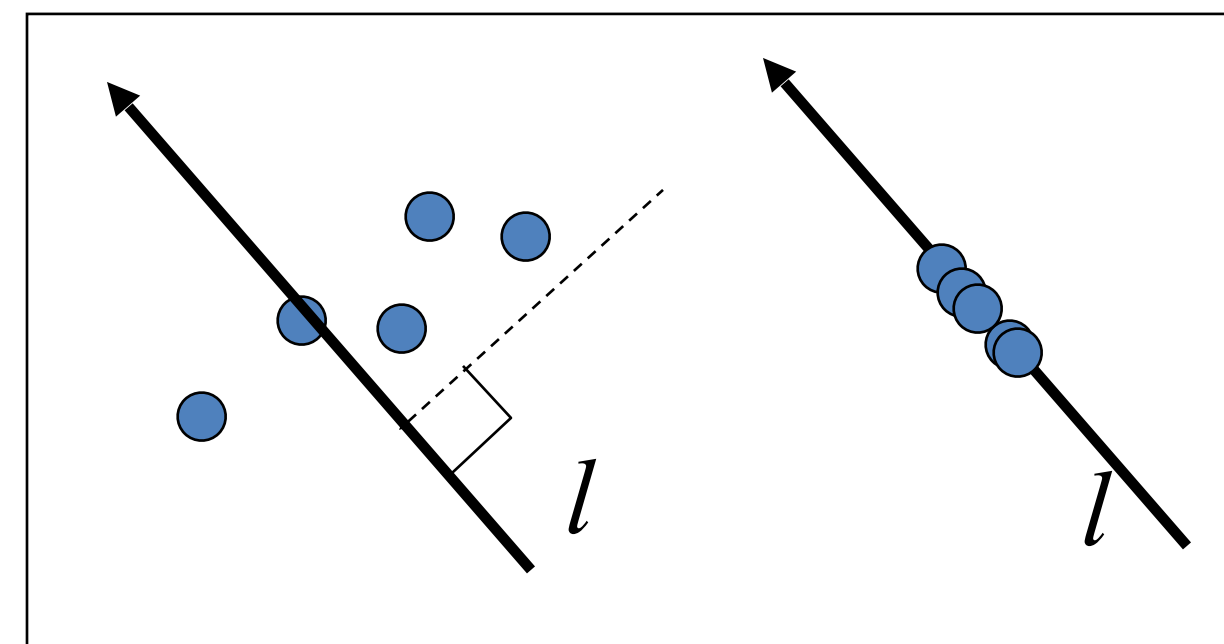
# What does the Scatter Matrix do?

- Let's look at a **line**  $l$  through the center of mass  $\mathbf{m}$  with direction vector  $\mathbf{v}$ , and project our points  $\mathbf{x}_i$  onto it. The **variance** of the **projected** points  $\mathbf{x}'_i$  is:

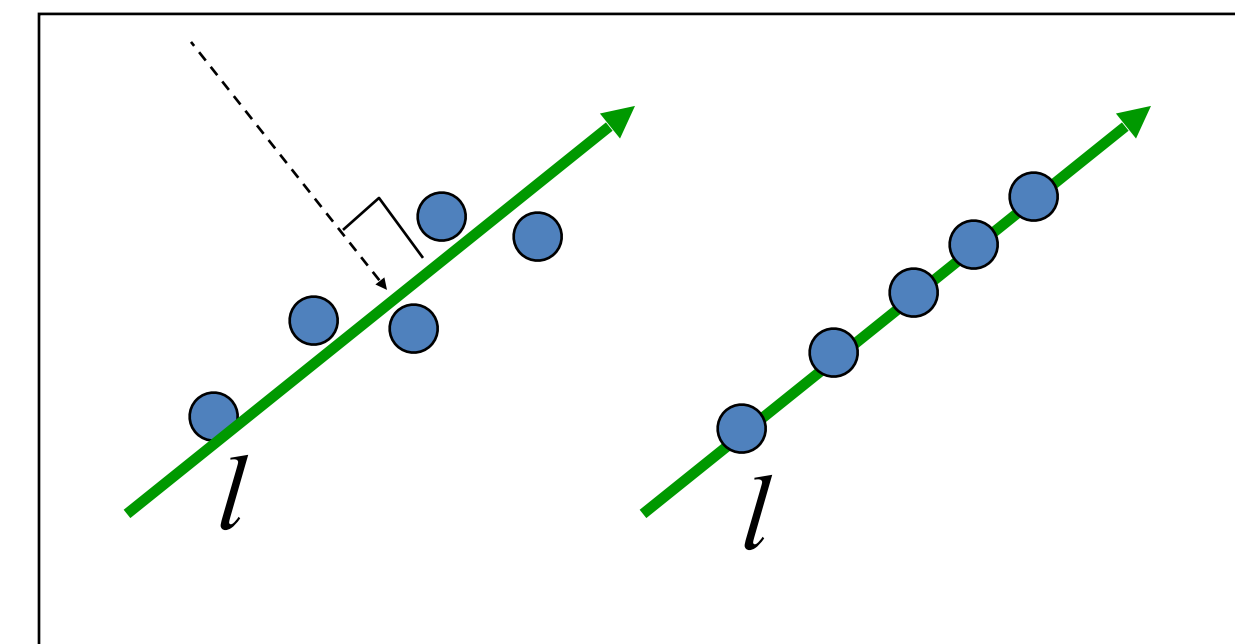
$$\begin{aligned} \text{var}(\mathbf{x}_1, \dots, \mathbf{x}_n; \mathbf{v}) &= \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}'_i - \mathbf{m}\|^2 = \\ &= \frac{1}{n} \sum_{i=1}^n \|(\mathbf{m} + \mathbf{v}^T \mathbf{y}_i) - \mathbf{m}\|^2 = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i^T \mathbf{v})^2 = \frac{1}{n} \mathbf{v}^T \mathbf{S} \mathbf{v} \end{aligned}$$



Original set



Small variance

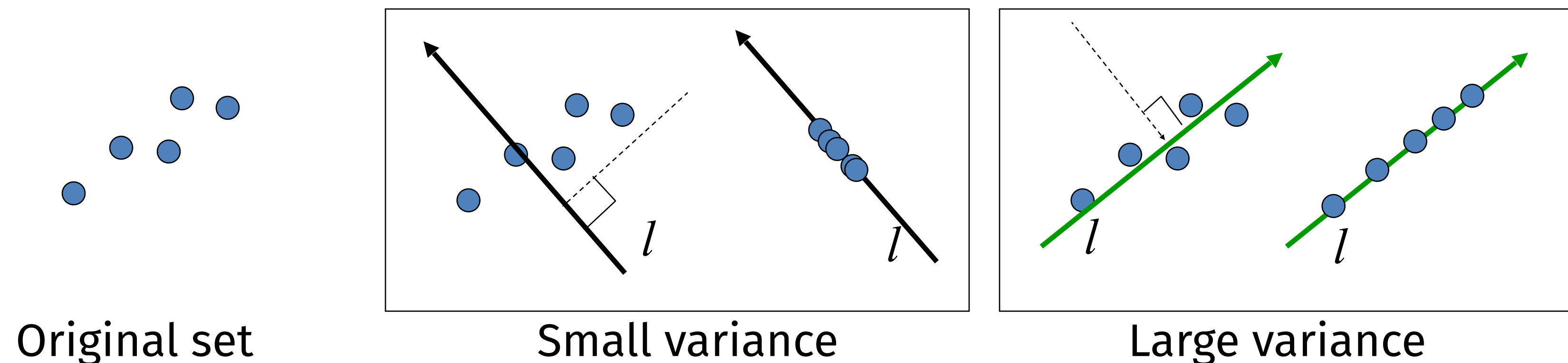
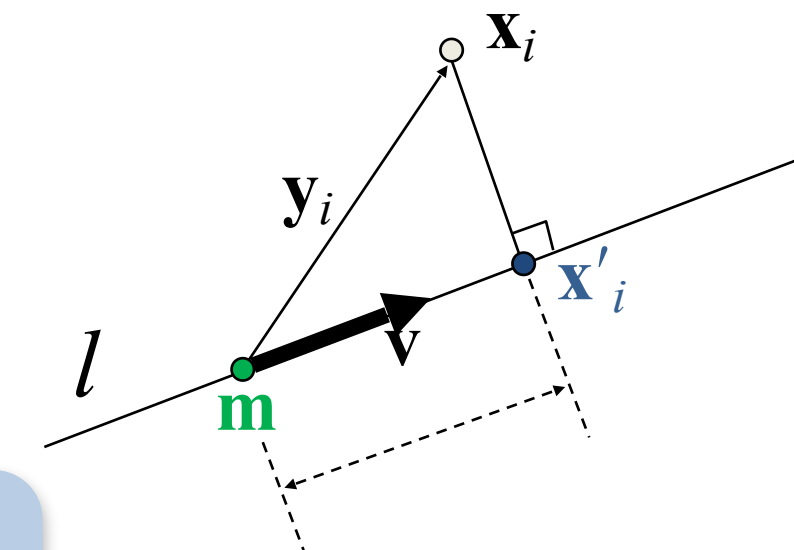


Large variance

# What does the Scatter Matrix do?

- The scatter matrix measures the variance of our data points along the direction  $\mathbf{v}$

$$\begin{aligned} \text{var}(\mathbf{x}_1, \dots, \mathbf{x}_n; \mathbf{v}) &= \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}'_i - \mathbf{m}\|^2 = \\ &= \frac{1}{n} \sum_{i=1}^n \|(\mathbf{m} + \mathbf{v}^T \mathbf{y}_i) - \mathbf{m}\|^2 = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i^T \mathbf{v})^2 = \frac{1}{n} \mathbf{v}^T \mathbf{S} \mathbf{v} \end{aligned}$$



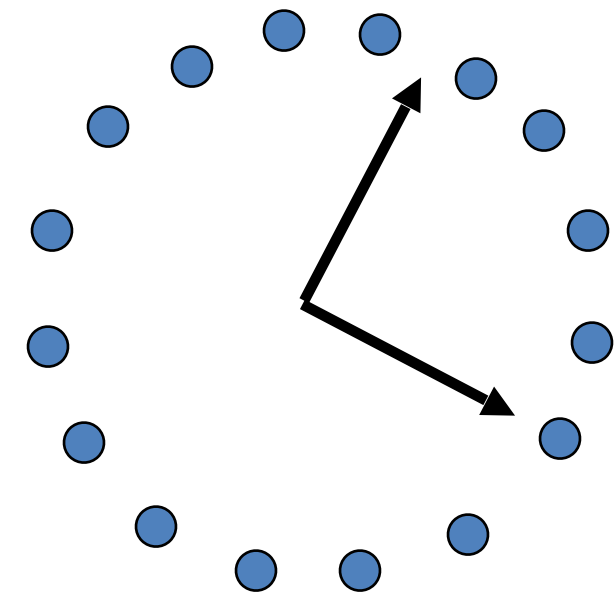


# Principal Components

- Eigenvectors of  $\mathbf{S}$  that correspond to **big** eigenvalues are the directions in which the data has strong components (= large variance).
- If the eigenvalues are more or less the same – there is no preferable direction.

$$\mathbf{S} = \mathbf{V} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix} \mathbf{V}^T$$

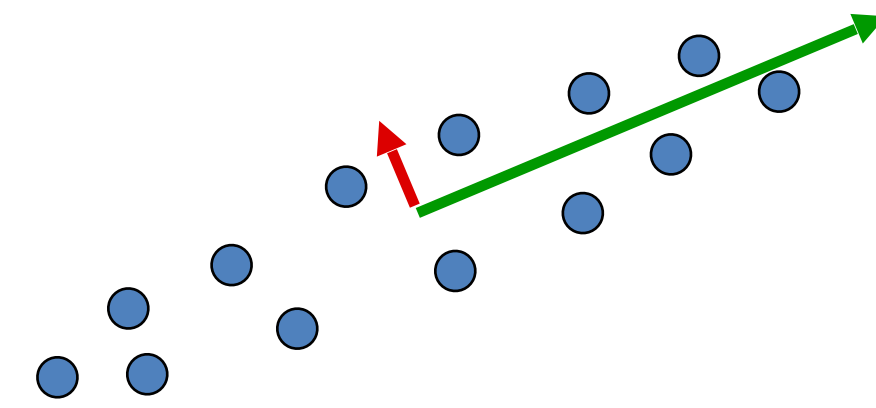
# Principal Components



- There's no preferable direction
- S looks like this:

$$\mathbf{S} = \mathbf{V} \begin{pmatrix} \lambda & \\ & \lambda \end{pmatrix} \mathbf{V}^T$$

- Any vector is an eigenvector



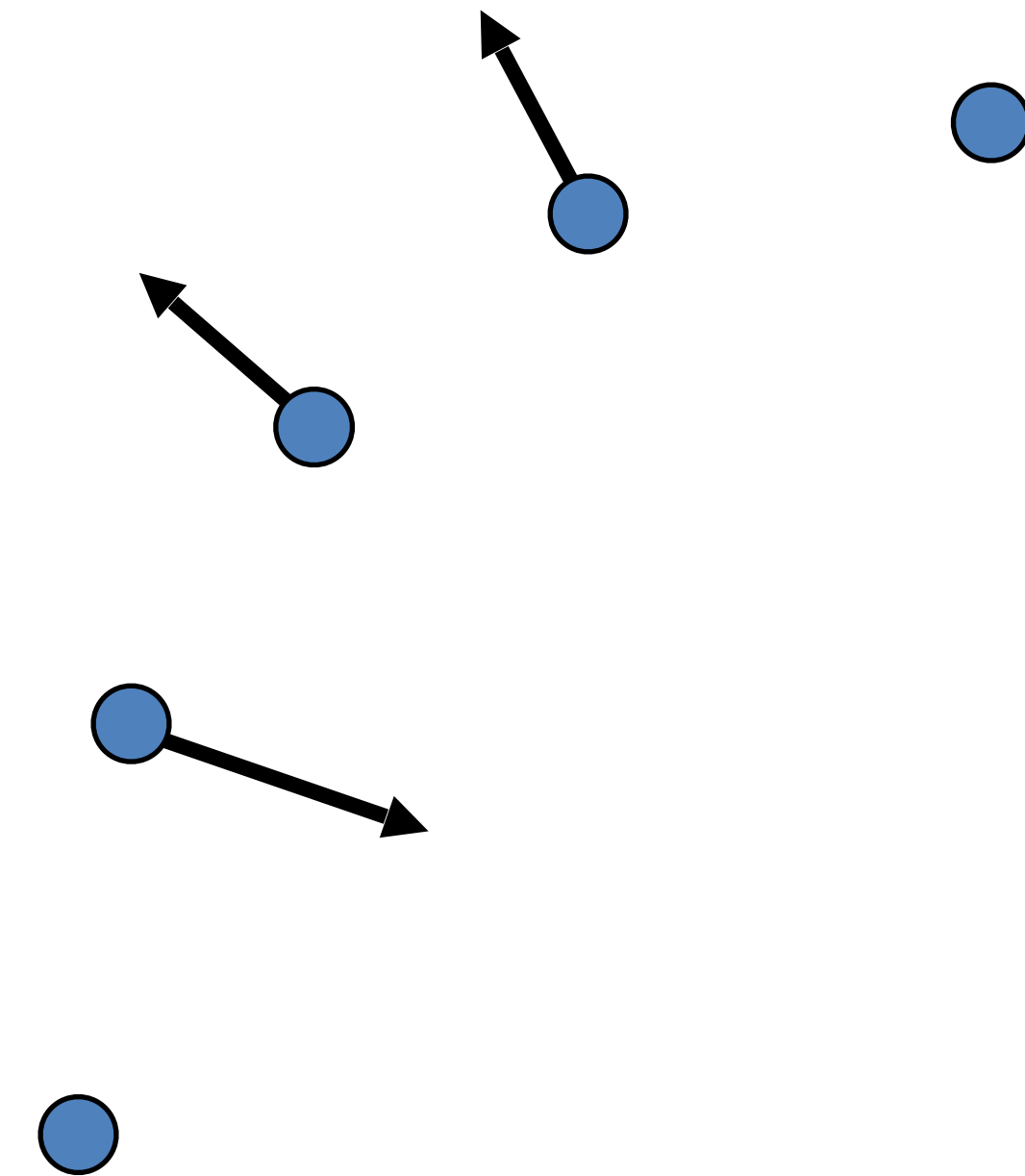
- There's a clear preferable direction
- S looks like this:

$$\mathbf{S} = \mathbf{V} \begin{pmatrix} \lambda & \\ & \mu \end{pmatrix} \mathbf{V}^T$$

- $\mu$  is close to zero, much smaller than  $\lambda$

# Normal Orientation

- PCA may return arbitrarily oriented eigenvectors
- Wish to orient consistently
- Neighboring points should have similar normals



# Normal Orientation

- Build graph connecting neighboring points
  - Edge  $(i,j)$  exists if  $\mathbf{x}_i \in \text{kNN}(\mathbf{x}_j)$  or  $\mathbf{x}_j \in \text{kNN}(\mathbf{x}_i)$
- Propagate normal orientation through graph
  - For neighbors  $\mathbf{x}_i, \mathbf{x}_j$  : Flip  $\mathbf{n}_j$  if  $\mathbf{n}_i^T \mathbf{n}_j < 0$
  - Fails at sharp edges/corners
- Propagate along “safe” paths (parallel tangent planes)
  - Minimum spanning tree with angle-based edge weights

$$w_{ij} = 1 - |\mathbf{n}_i^T \mathbf{n}_j|$$

“Surface reconstruction from unorganized points”, Hoppe et al., SIGGRAPH 1992  
<http://research.microsoft.com/en-us/um/people/hoppe/recon.pdf>

# Thank you