Lecture 5

Edge Detection 2: Canny

Least squares estimation 1: line fitting

Recall… Edge Detection

Two issues to deal with:

1. Edges can have any orientation.

2. Edges should be thin (preferably one pixel wide).

Marr and Hildreth's method handles these issues.

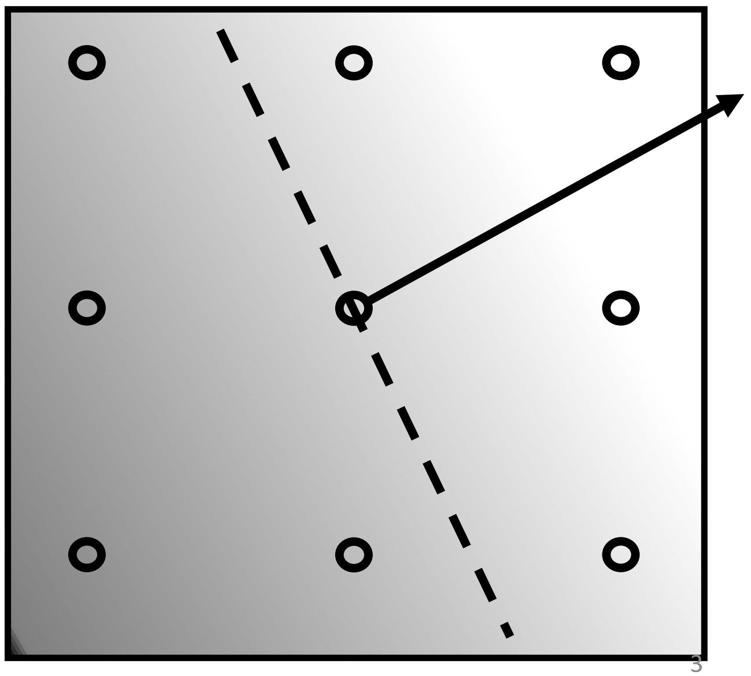
We next discuss another classic method: the Canny edge detector (1986).

Canny Edge Detection (1986)

Step 1:   Compute gradient at each pixel.

$$\nabla G(x,y,\sigma) * I(x,y)$$

$$\equiv \left(\frac{\partial}{\partial x} G(x,y,\sigma) * I(x,y), \qquad \frac{\partial}{\partial y} G(x,y,\sigma) * I(x,y) \right)$$



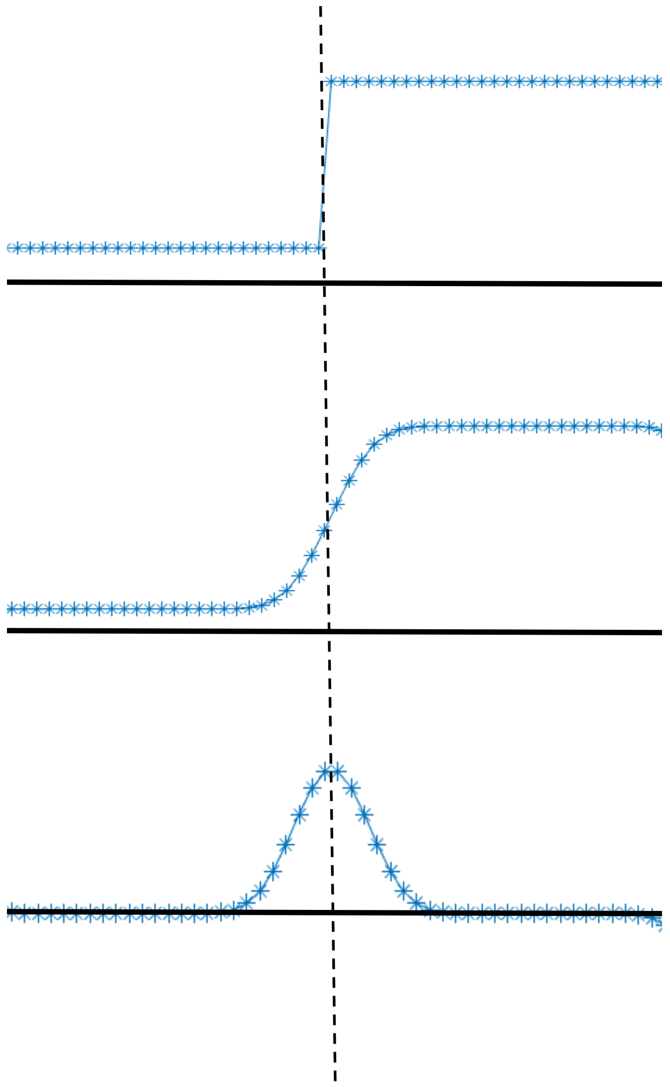
Mark a pixel as a candidate edge if its gradient magnitude is greater than some threshold τ_{high}.

For a blurred 1D step edge, the edge location is at the maxima of first derivative. However, there may be neighboring points that also have large first derivative so thresholding is not sufficient.

$$I(x)$$

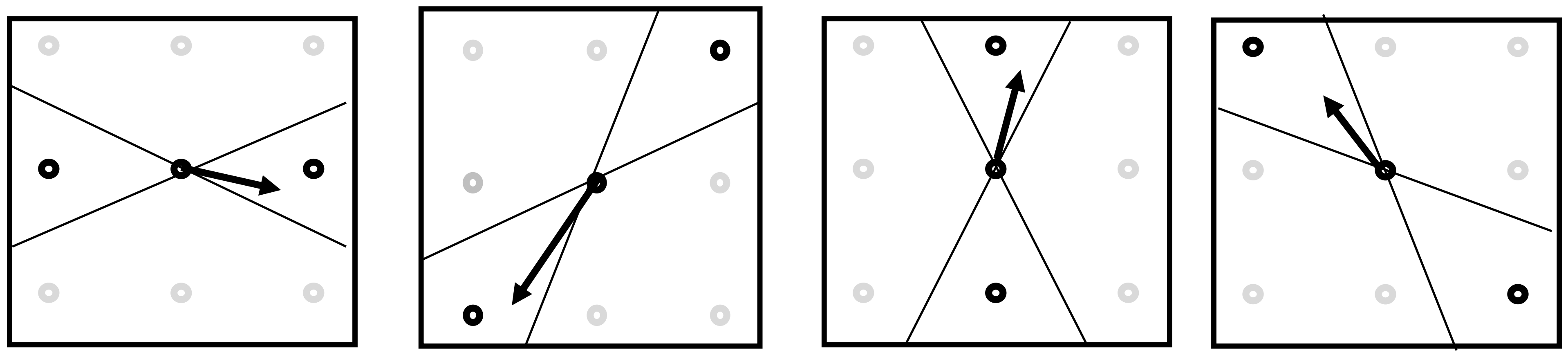
$$G(x,\sigma) * I(x)$$

$$\frac{d\, G(x,\sigma) * \ I(x)}{dx}$$



Canny Edge Detection

Step 2: “non-maximum suppression”

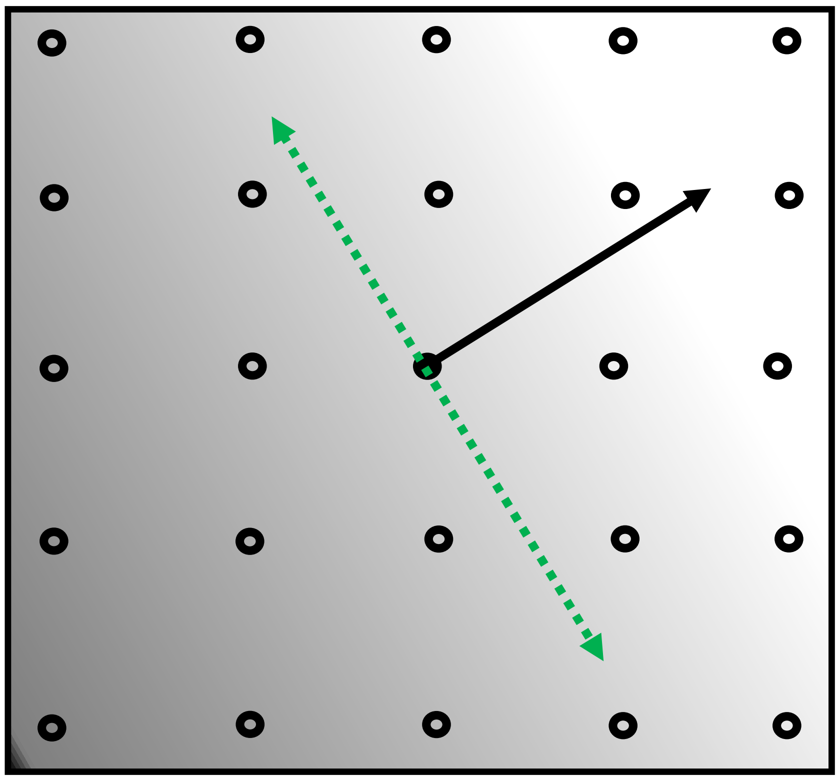


Compare the gradient magnitude of an edge from step 1 to the gradient magnitude of the neighbors who are *nearest to the direction of that gradient*. Eliminate edge pixels if the magnitude of their gradient is *not* a local maximum relative to the neighbors.

(Here only a 3x3 neighborhood is shown but a bigger neighborhood could be used.)

Canny Edge Detection

Step 3: "hysteresis" thresholding (details omitted)

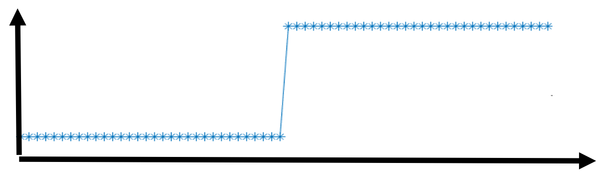


For each edge pixel that survives step 2, search neighbors in two opposite directions perpendicular to the gradient and find a chain of pixels whose gradient magnitudes are all greater than a threshold τ_{low} ($\tau_{low} < \tau_{high}$).
Consider the pixels in such chains to be edge pixels too.

J. Canny: "A Computational Approach to Edge Detection"

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.420.3300&rep=rep1&type=pdf

Canny also introduced theoretical criteria for a filter to detect *1D step edges in the presence of noise.*



1. An edge detector should have a much larger response to the edge than it should to noise.

2. An edge detector should accurately estimate the *location* of edges.

He showed mathematically how varying the sigma of the Gaussian smoothing trades off (1) versus (2): increasing sigma improves the ratio of response of edge to response of noise, but reduces localization accuracy.

Lecture 5

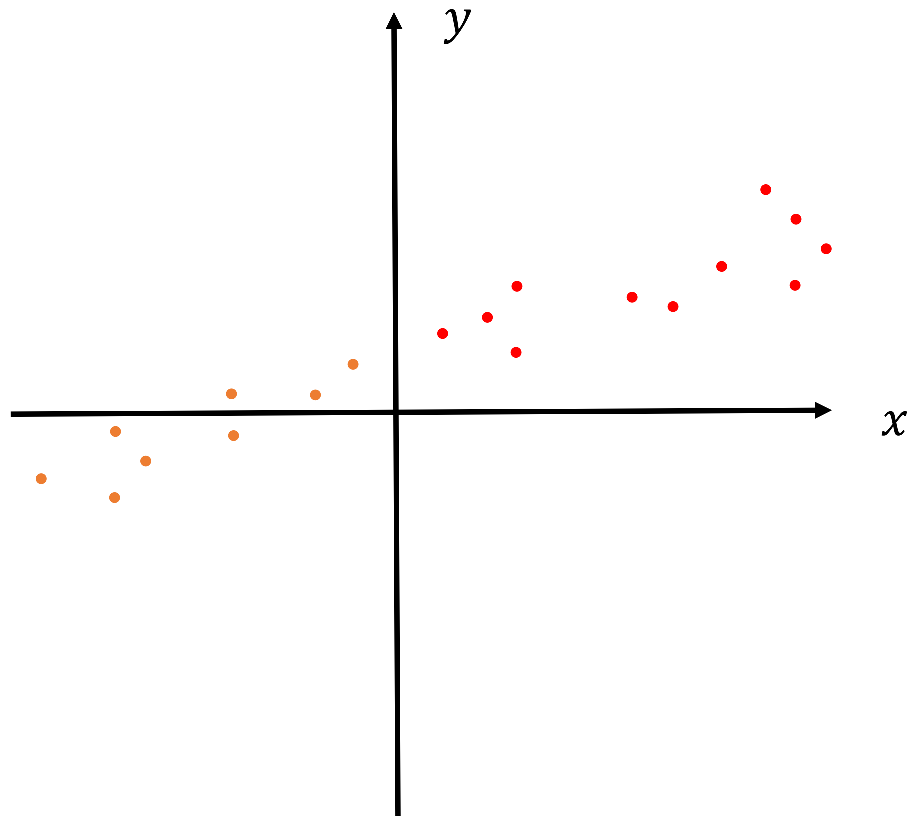
Edge Detection 2: Canny

Least squares estimation:
lines and vanishing points

Example Problem 1

How to fit a line to a set of points (x_i, y_i)?

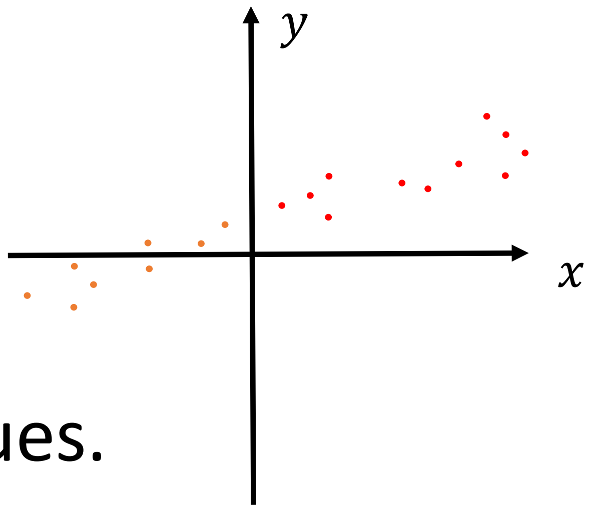
These could be locations of edges (with orientation ignored).



Least squares: version 1 (linear regression)

Model is:

$$y_i = m\, x_i + b + n_i$$



where n_i is additive noise in the y_i values.

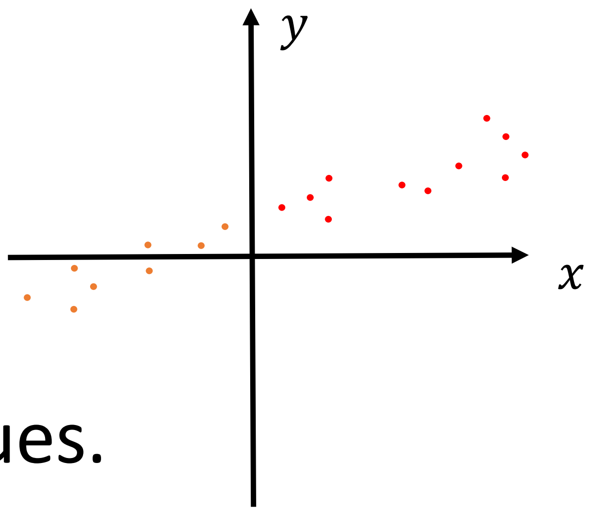
We want to solve for m and b. How ?

Least squares: version 1 (linear regression)

Model is:

$$y_i = m\, x_i + b + \, n_i$$

where n_i is additive noise in the y_i values.



We solve for:

$$\underset{m,\,b}{\operatorname{argmin}} \sum_i (y_i - (m\, x_i + b))^2$$

"Why least *squares*?"

Two reasons:

1. It is mathematically convenient because it allows one to solve minimization problems (e.g. find m, b) using linear algebra.

2. If the noise n_i has a Gaussian probability density, then one can show that minimizing the squared errors gives the best estimate of m, b in a "maximum likelihood" sense.

[ASIDE: I am omitting technical details about maximum likelihood here because it would be distracting for those who don't know what it is. The term "likelihood" refers to a particular is a conditional probability. The maximum likelihood method maximizes that conditional probability.]

Take derivative of $\sum_i (y_i - (m\,x_i + b))^2$ w.r.t. b and set to 0:

$$\sum_i 2\,(y_i - (m\,x_i + b)) \;=\; 0$$

Take derivative of $\sum_i (y_i - (m\,x_i + b))^2$ w.r.t. m and set to 0:

$$\sum_i 2\,(y_i - (m\,x_i + b))\; x_i \;=\; 0$$

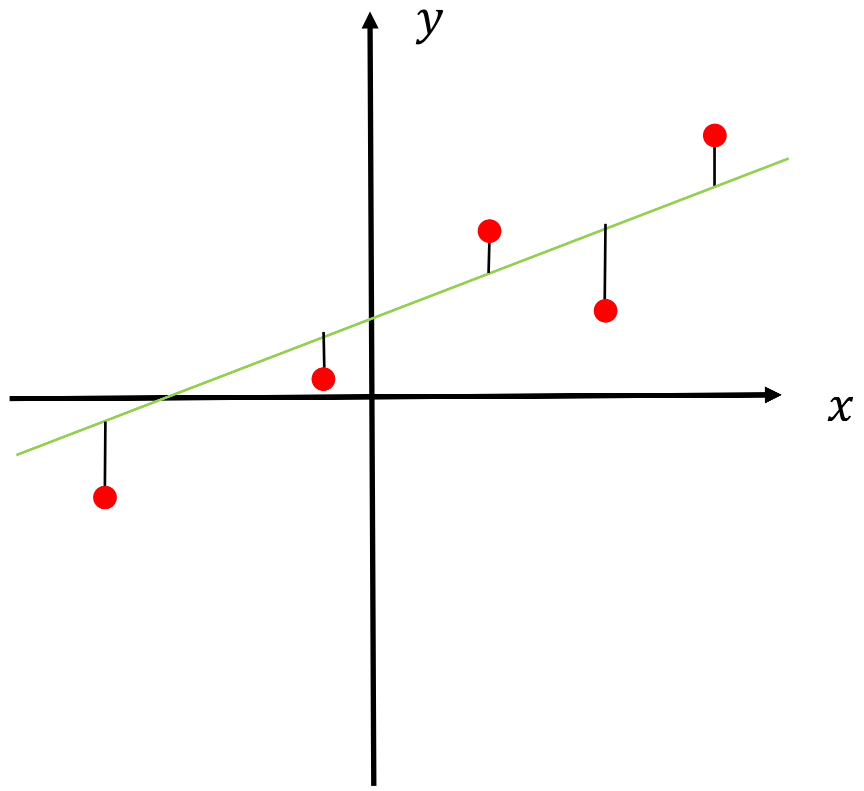
So we have two linear equations with two unknowns and we can solve for m, b.

[This method goes back to Gauss in early 1800's.]

$$\begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} x_i \, y_i \\ \sum_{i=1}^{N} y_i \end{bmatrix}$$

Version 1: linear regression

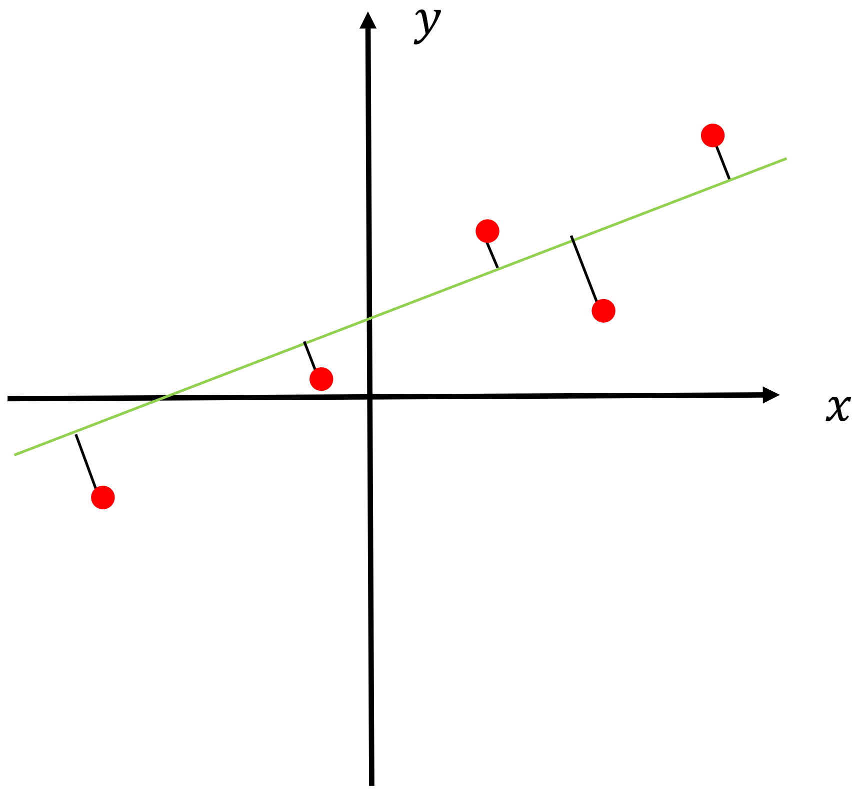
Error is distance to line in y direction only.



$$y = mx + b$$

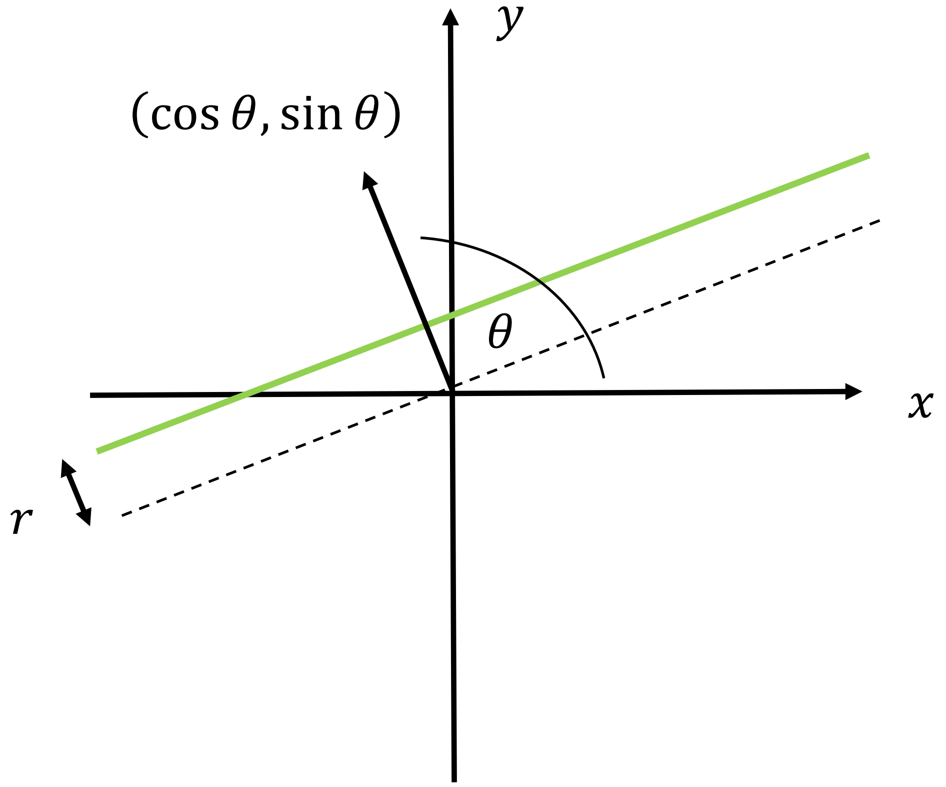
Version 2: "total least squares"

Error is distance perpendicular to line.



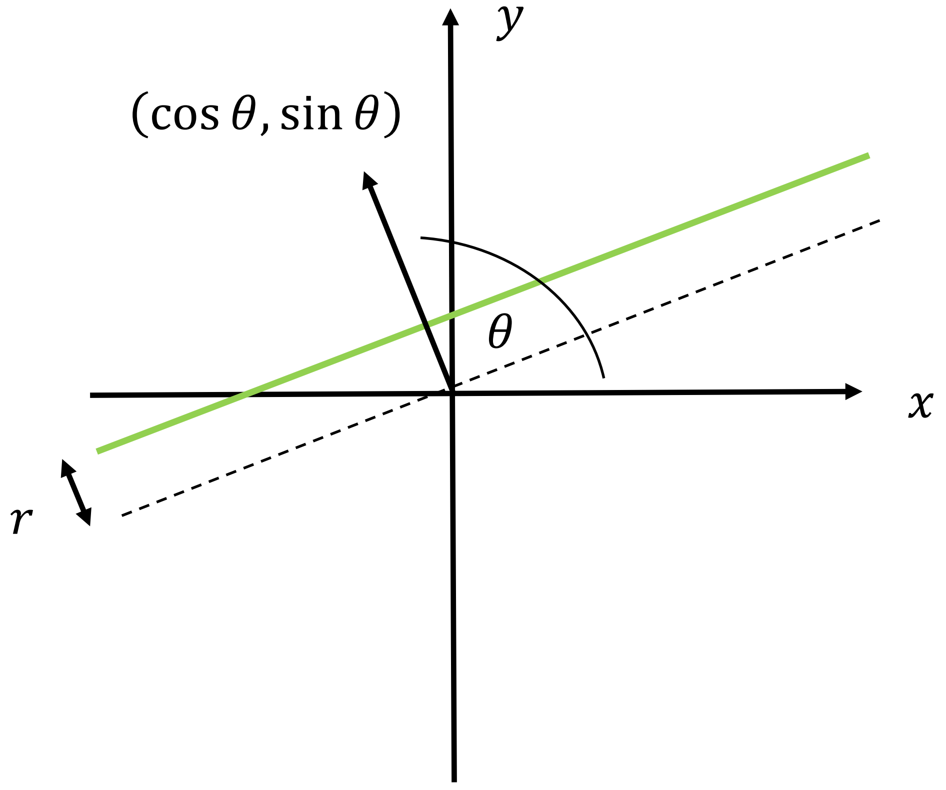
We need a different line representation for perpendicular distances.

A line can be represented using an angle θ in [0, 360) degrees, and a non-negative perpendicular distance r of the line away from the origin.



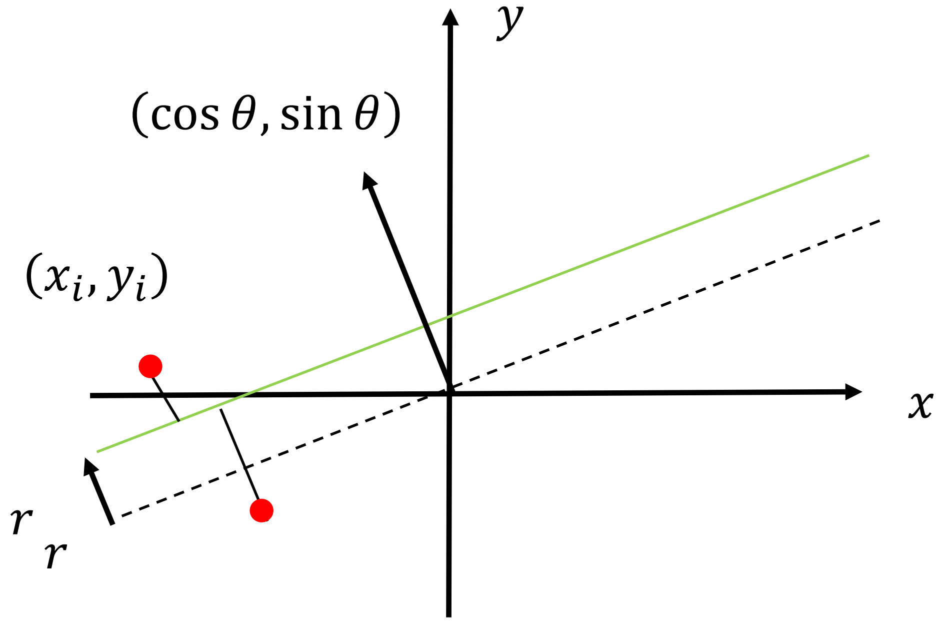
How ?

A line can be represented using an angle θ in [0, 360) degrees, and a non-negative perpendicular distance r of the line away from the origin.

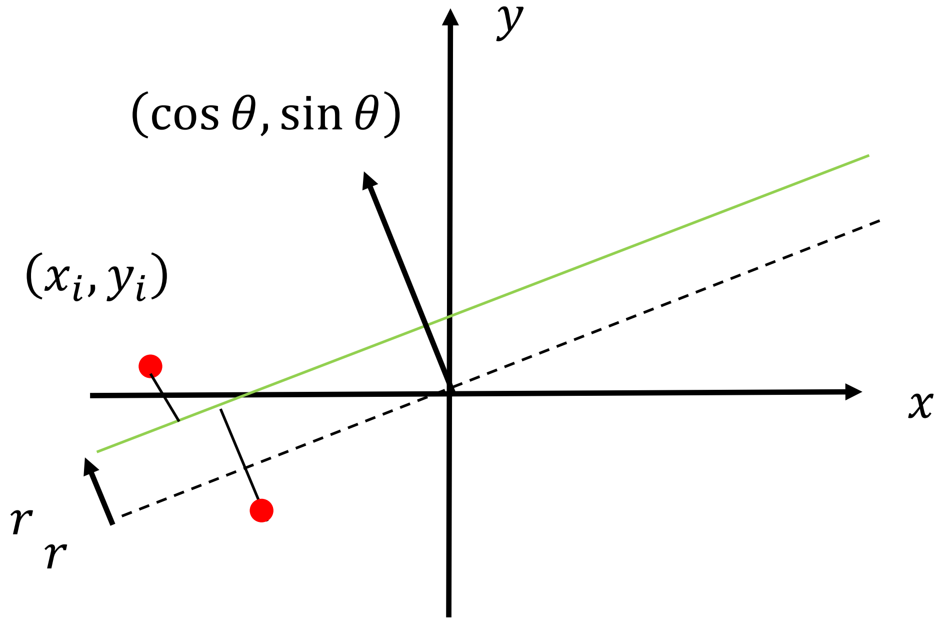


$$x\ \cos\theta + y\ \sin\theta\ = r$$

What is the distance from any point (x_i, y_i) to the line ?



What is the distance from any point (x_i, y_i) to the line ?



The distance from (x_i, y_i) to the dashed line through the origin is

$$|\ x_i \cos\theta + y_i \sin\theta\ |.$$

The distance from (x_i, y_i) to the model line with parameters (θ, r) is

$$|\ x_i \cos\theta + y_i \sin\theta\ \ -\ r\ \ |$$

Let n_i be a point's position noise *in the direction perpendicular to the line*.

$$n_i \;=\; x_i \cos\theta + y_i \sin\theta \;-\; r\,.$$

Solve for:

$$\underset{\theta,\, r}{\operatorname{argmin}} \sum_i (\, x_i \cos\theta + y_i \sin\theta \;-\; r\,)^2$$

Solve for:

$$\underset{\theta, r}{\operatorname{argmin}} \sum_i (x_i \cos\theta + y_i \sin\theta \; - \; r)^2$$

Take derivative with respect to r and set it to 0 gives:

$$0 = \sum_i (x_i \cos\theta + y_i \sin\theta \; - \; r)$$

So, $\quad 0 = \bar{x} \cos\theta + \bar{y} \sin\theta \; - r$

i.e. $\quad (\bar{x}, \bar{y}) = \; \frac{1}{N} \sum_{i=1}^{N} \; (x_i, y_i)$ lies on the best fitting line.

Now we can solve for θ. Using the result on the previous slide, i.e. substituting for r, we want to find:

$$\underset{\theta}{\operatorname{argmin}} \sum_{i} (\, (x_i - \bar{x}) \cos\theta + (y_i - \bar{y}) \sin\theta \,)^2$$

How do we find this θ using linear algebra?

[Taking derivative with respect to θ and set to 0 is awkward.]

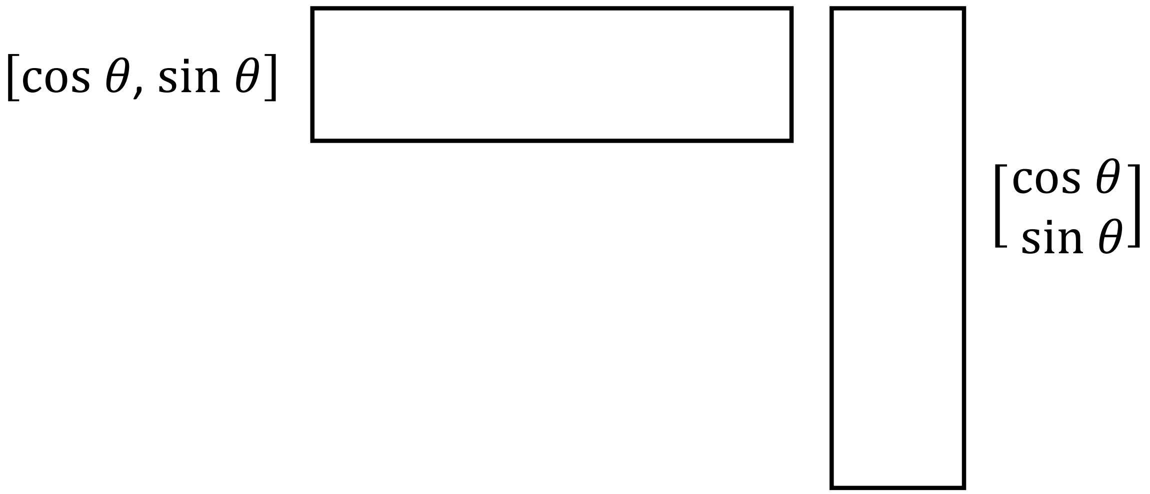
We can write:

$$\sum_i ((x_i - \bar{x}) \cos\theta + (y_i - \bar{y}) \sin\theta)^2$$

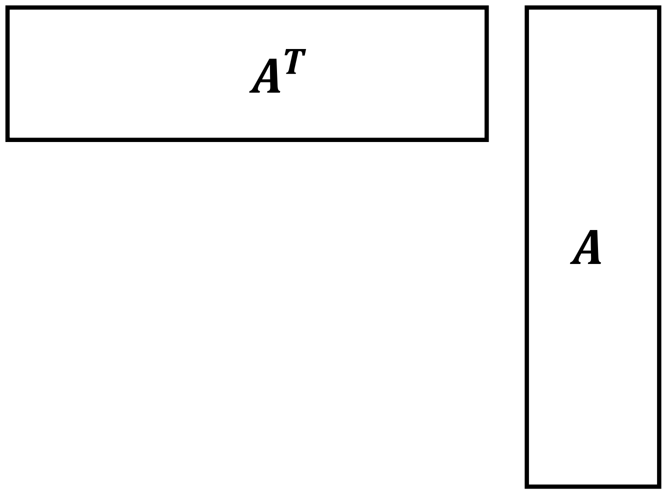
as a matrix product:

$$[\cos\theta, \sin\theta] \; \boldsymbol{A^T A} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

where $\boldsymbol{A}$ is an $N \; x \; 2$ matrix with rows $(x_i - \bar{x}\,, \;\; y_i - \bar{y})$ for $i = 1 \; to \; N$.


$[\cos\theta, \sin\theta]$
$\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$

$\boldsymbol{A}$ is an $N\ x\ 2$ matrix with rows $(x_i - \bar{x}\,,\ \ y_i\ -\ \bar{y})$ for $i = 1\ to\ N$.



and $\boldsymbol{A^T A}$ is a 2x2 matrix:

$$\begin{bmatrix} \sum_{i=1}^{N} (x_i - \bar{x}\,)^2 & \sum_{i=1}^{N} (x_i - \bar{x}\,)\ (y_i\ -\ \bar{y}\,) \\ \sum_{i=1}^{N} (x_i - \bar{x}\,)\ (y_i\ -\ \bar{y}\,) & \sum_{i=1}^{N} (y_i - \bar{y}\,)^2 \end{bmatrix}$$

Note that:

$$[\cos\theta, \sin\theta]\ \boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}\begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix} = \left\| \boldsymbol{A}\begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix} \right\|^2 \geq 0$$

We want to find θ that minimizes this quantity. How ?

Note that:

$$[\cos\theta, \sin\theta] \ \boldsymbol{A^T A} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = \left\| \boldsymbol{A} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \right\|^2 \geq 0$$

We want to find θ that minimizes this quantity. How ?

Compute the eigenvectors and eigenvalues of $\boldsymbol{A^T A}$.

Take the unit eigenvector which has the smaller eigenvalue.

Take the θ such that $[\cos\theta, \sin\theta]^T$ is this unit eigenvector.

Later in the course, I will present a more general formulation.