3D – Model Fitting
Expectation Maximization

Derek Nowrouzezahrai
derek@cim.mcgill.ca

What if…

We discussed and related MLE, MAP and LLS estimators

- showed a few examples of how to derive such estimators:
  - simple linear models
  - Normal distribution on the noise and parameter priors
    - 0- and non-zero mean
    - same vs. different variances

What if… things get complicated?

What happens when we deviate from these simple cases?

- more complicated models
- more complicated noise/prior distributions

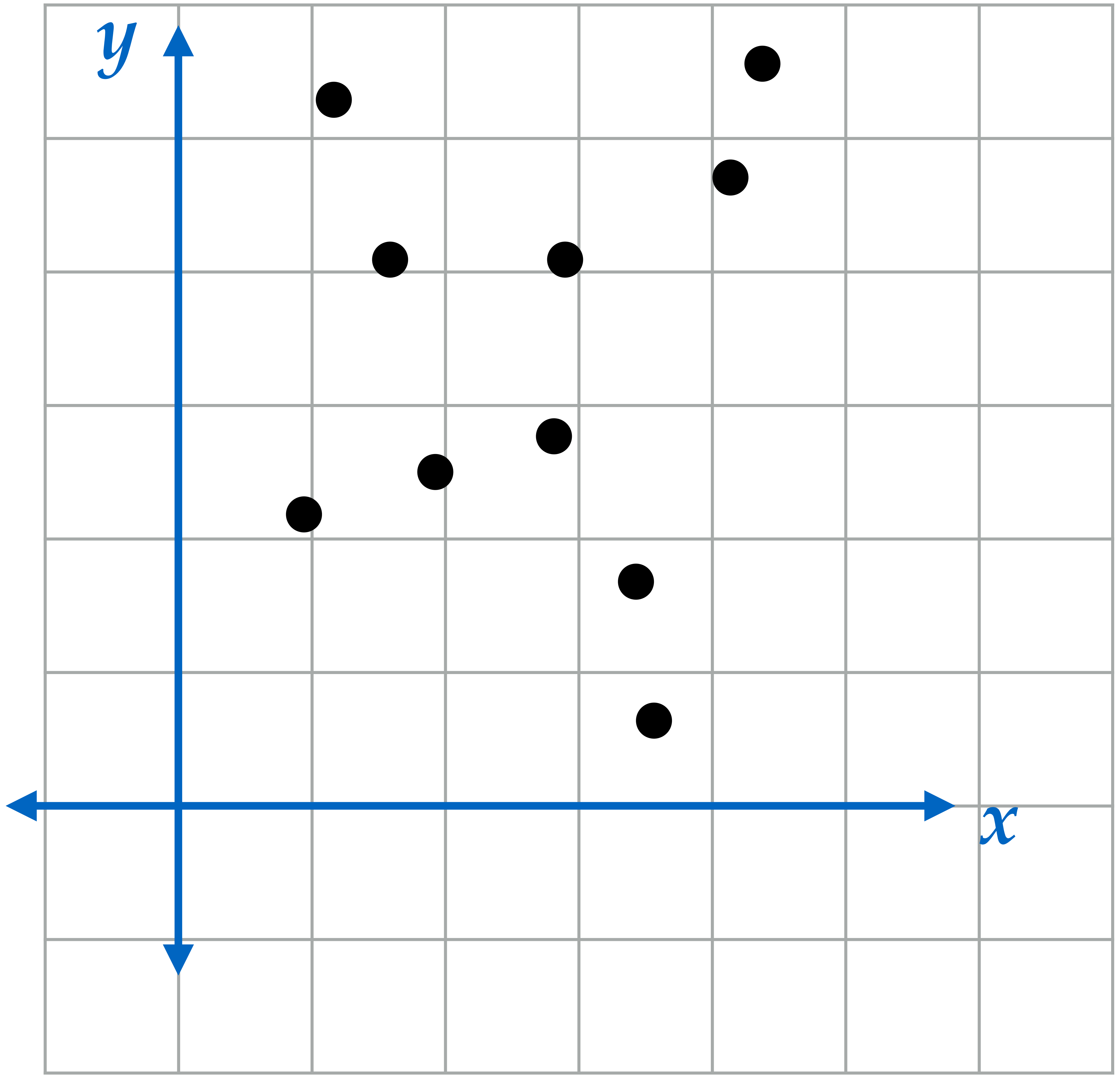
Still want the same guarantees as MLE or MAP

- maximizing the likelihood or posterior

Not a problem! Just work through the derivation…

- *"I did! But I couldn't solve for an analytic expression…"*

Example – Mixture Models



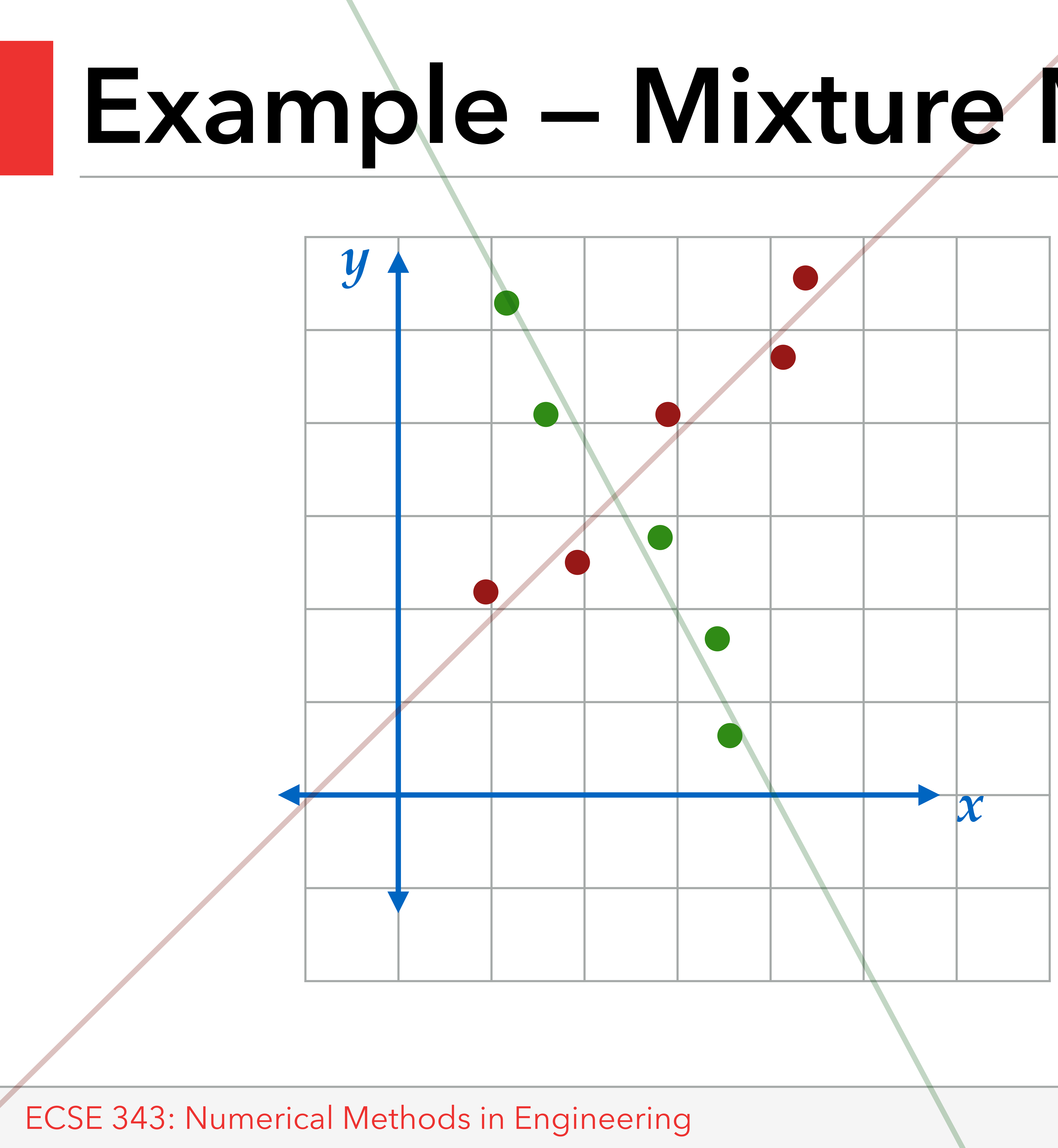
Given this dataset, if I ask you to fit a simple model (e.g., line or polynomial) you should be comfortable with:

- LS, WLS, MLE, MAP

What if your model is a *little bit* more complicated?

- specifically, sufficiently more complicated that it doesn't fit into the templates we've seen so far…
  - e.g., can't formulate it as a LS

Example – Mixture Models



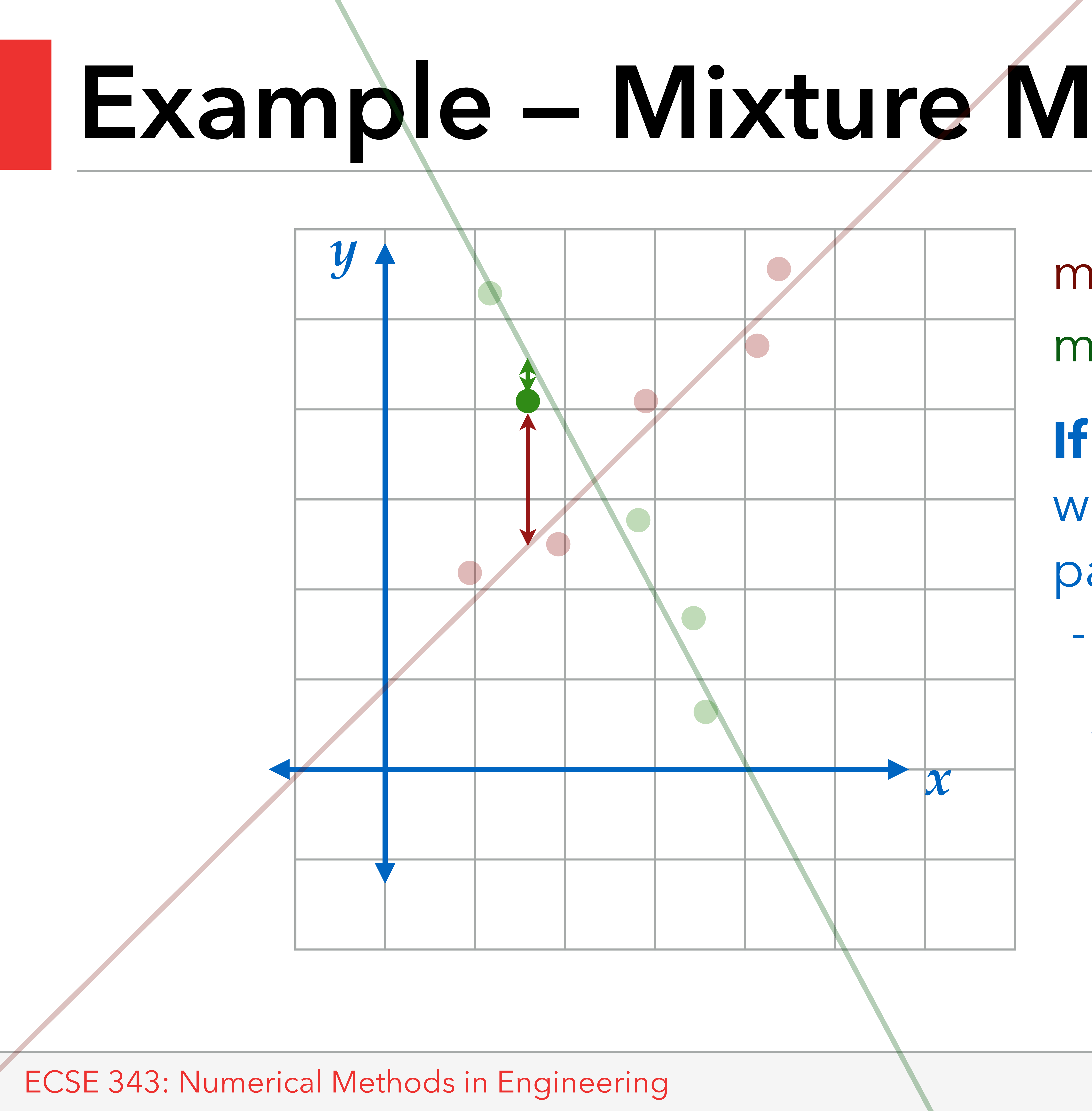
model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

If we know which data belong to which model, then estimating model parameters (a^1/b^1 and a^2/b^2) is easy

- LS, WLS, MLE, MAP

Example – Mixture Models



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

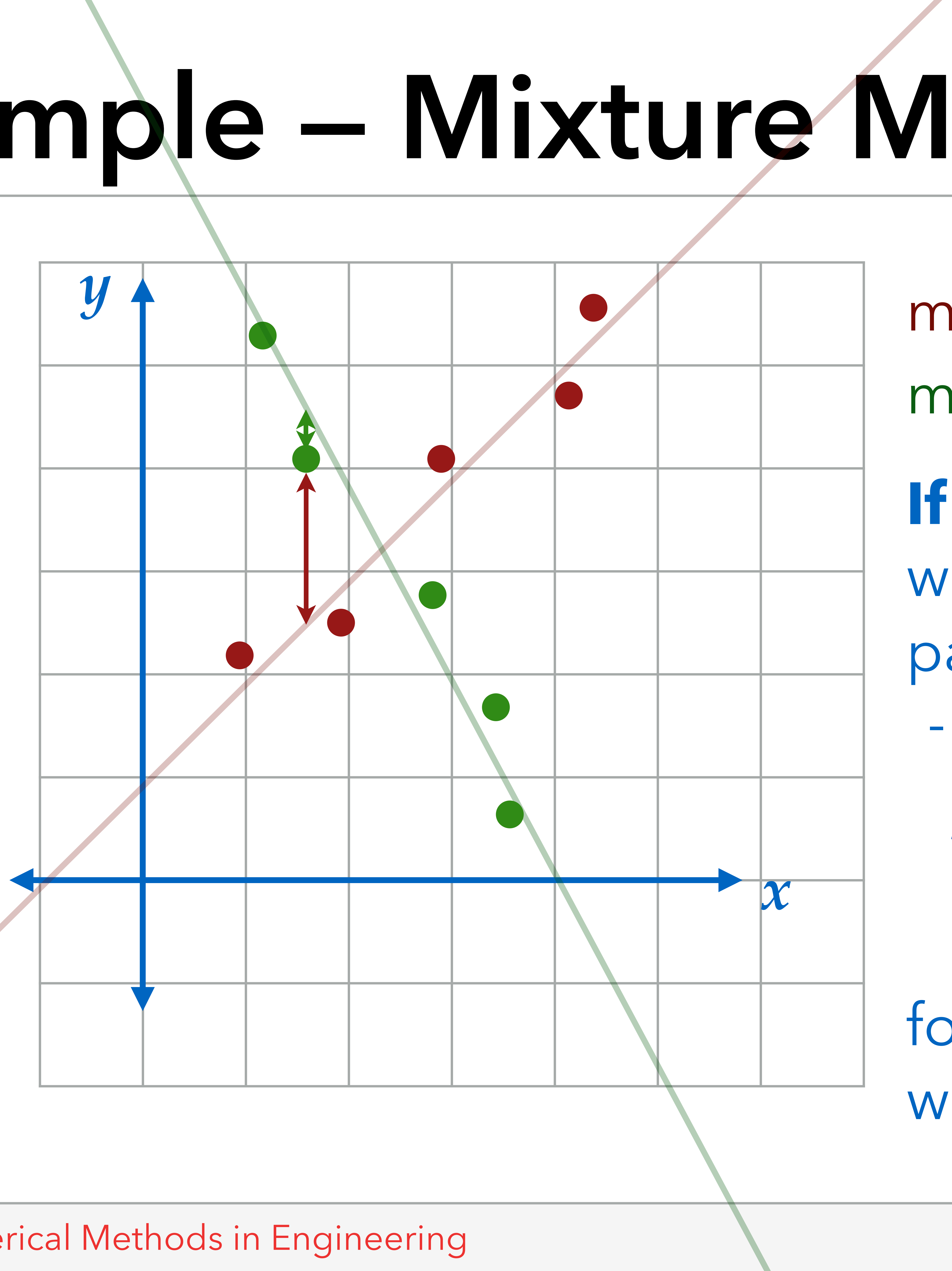
If we know which data belong to which model, then estimating model parameters (a^1/b^1 and a^2/b^2) is easy

- LS, WLS, MLE, MAP

$$r_i^k = |a^k x_i + b^k - y_i|$$

$$k = 1, 2$$

Example – Mixture Models



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

If we know which data belong to which model, then estimating model parameters (a^1/b^1 and a^2/b^2) is easy

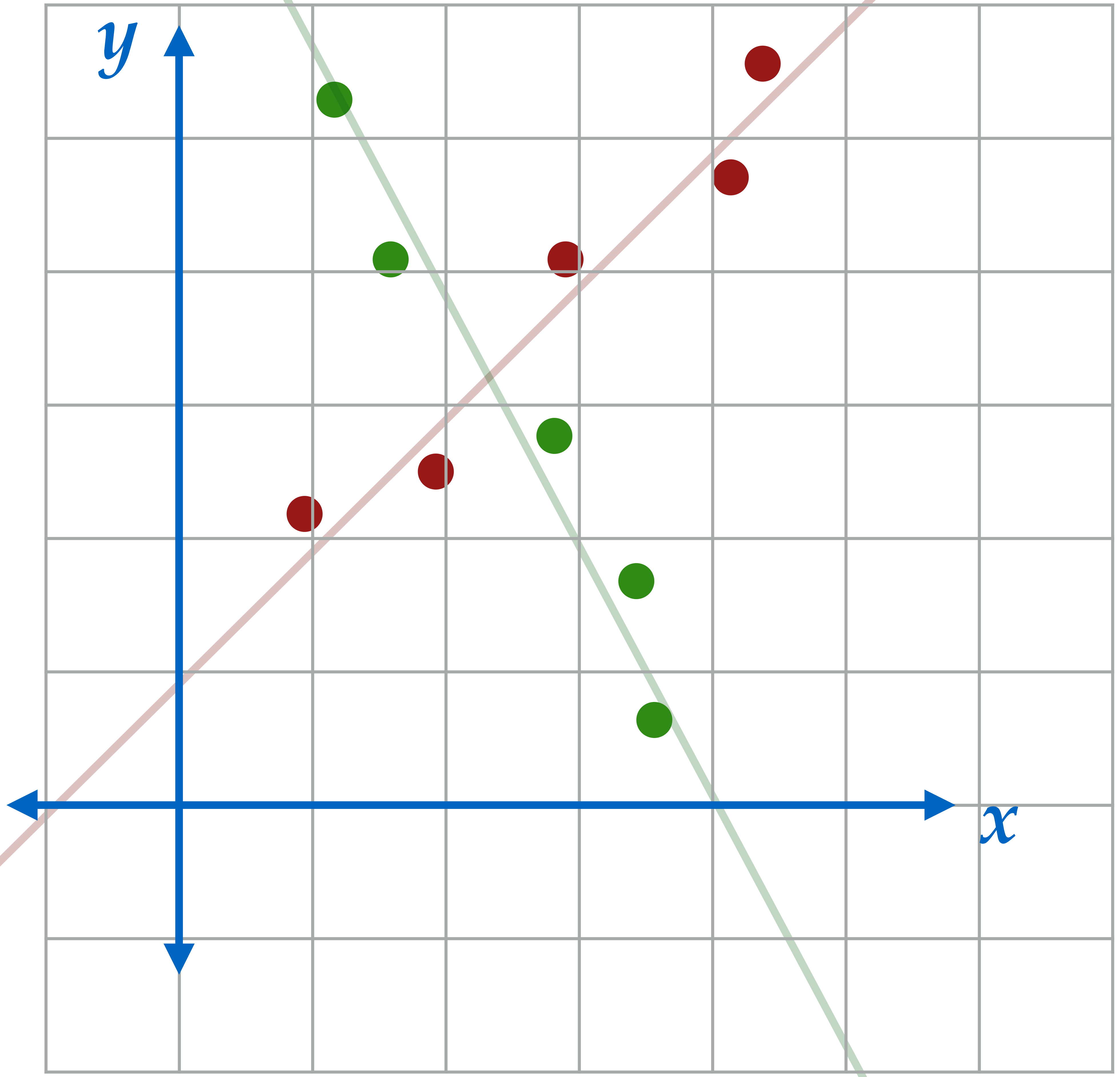
- LS, WLS, MLE, MAP

$$r_i^k = |a^k x_i + b^k - y_i|$$

$$k = 1, 2$$

for each (x_i, y_i), the model k is that which minimizes the residual r_i^k

Expectation-Maximization (EM)



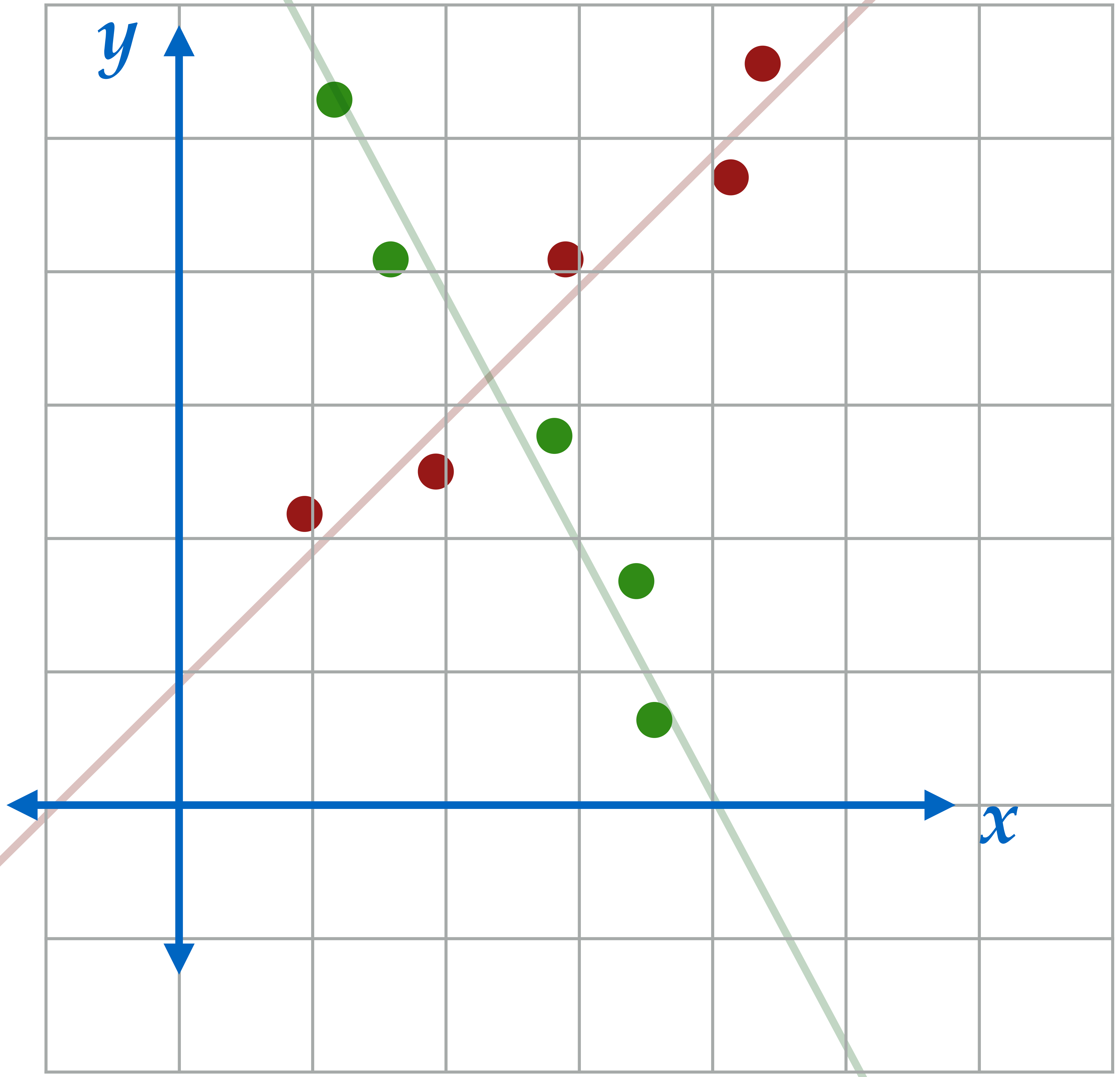
model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

The EM algorithm is an algorithm that iteratively estimates model parameters

- particularly useful when you cannot derive an analytic expression for an estimator (whether MLE or MAP)

Expectation-Maximization (EM)



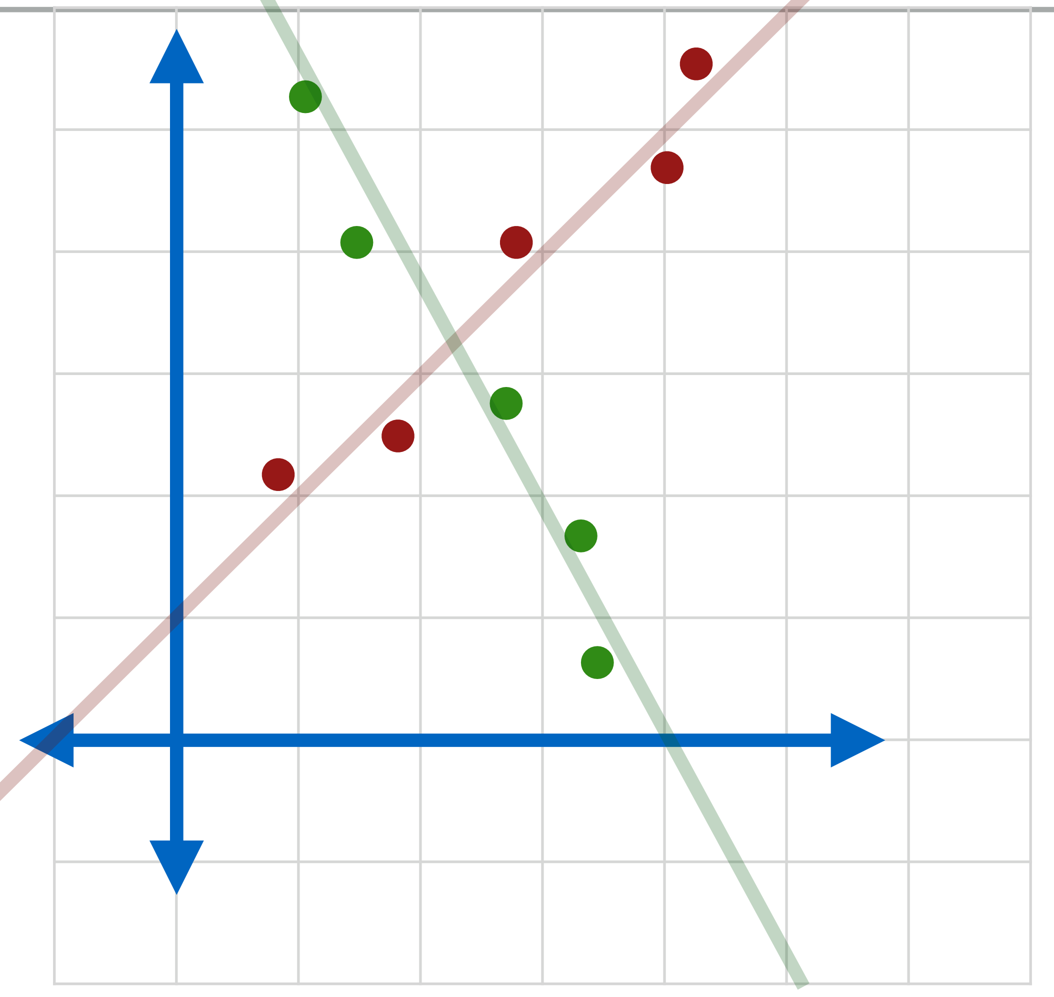
model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

In this example, EM will allow us to estimate the **mixture assignment** *and* **model parameters** that maximize the likelihood or posterior – subject to any models we have for the noise and parameter priors

$$EM = E + M$$

E-step



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

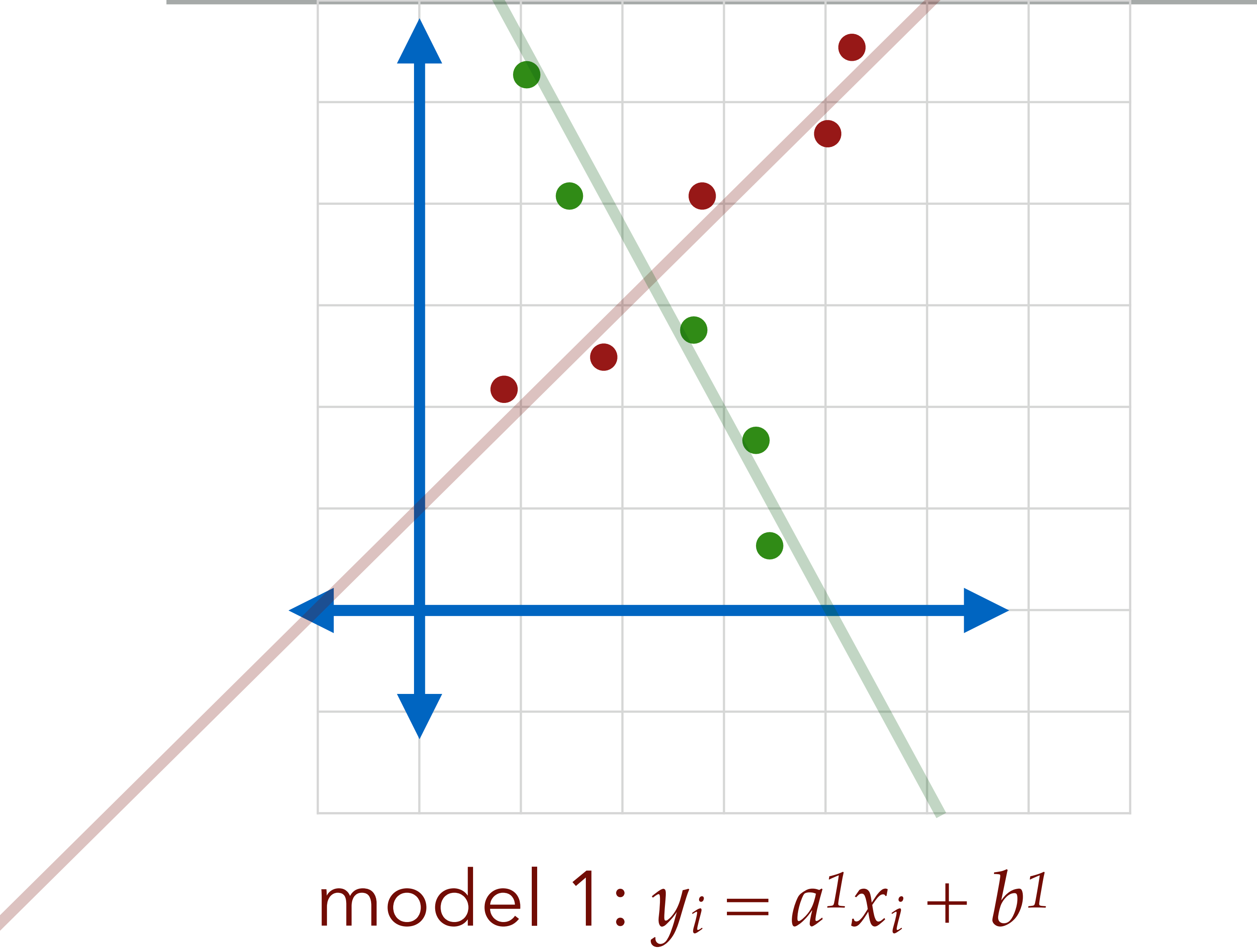
E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

For each data point i, and for each model k compute the residual, e.g., $r_i^k = |a^k x_i + b^k - y_i|$

Given the residual for each model, what is the probability $P(i \in M_k | r_i^k)$ that a data point (x_i, y_i) belongs to model k?

$$P(i \in M_k | r_i^k) = \frac{P(r_i^k | i \in M_k) P(i \in M_k)}{P(r_i^k)}$$

E-step



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

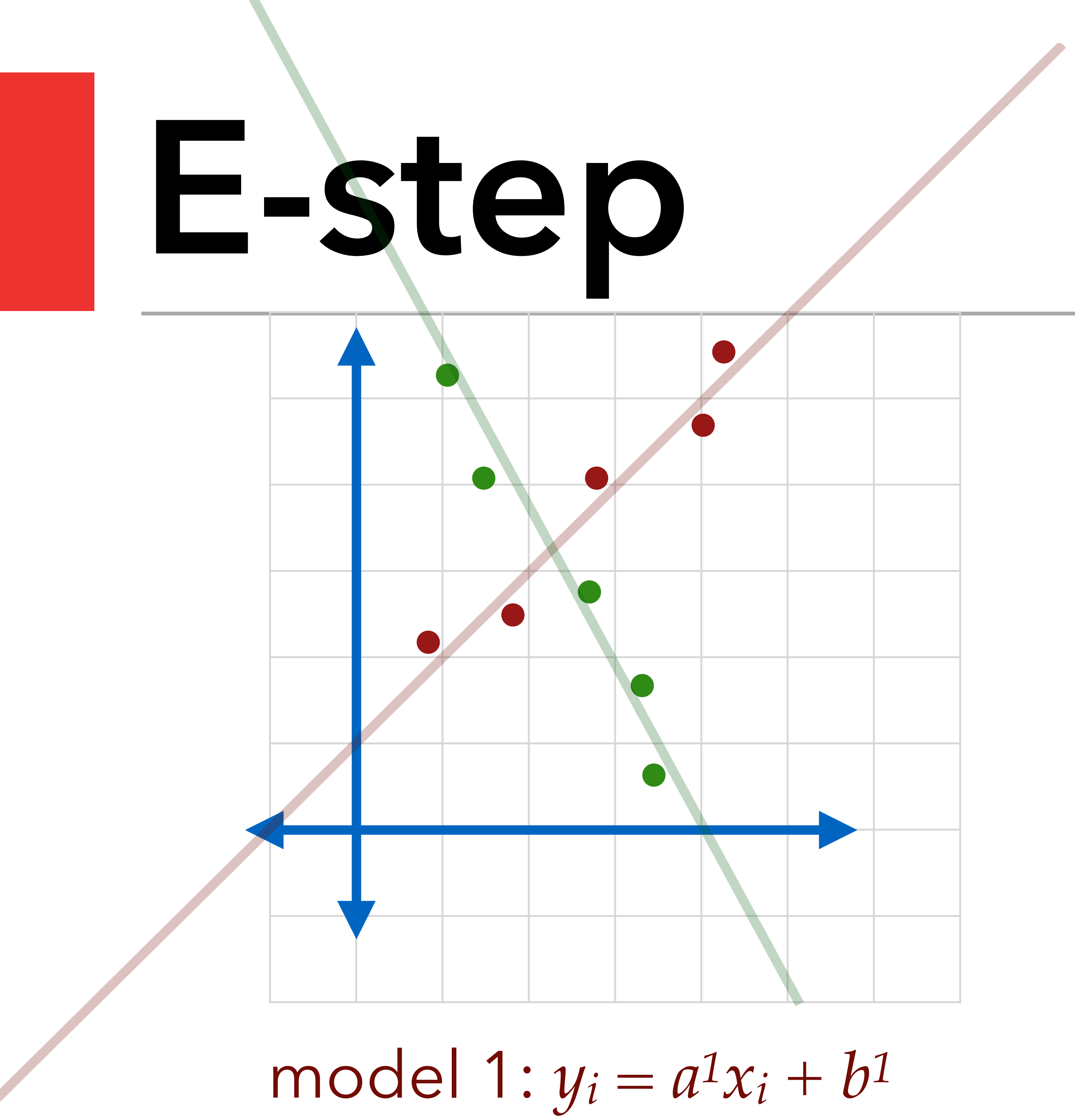
Probability that a data point (x_i, y_i) belongs to model k:

$$P(i \in M_k | r_i^k) = \frac{P(r_i^k | i \in M_k) P(i \in M_k)}{P(r_i^k)}$$

According to the law of total probability:

$$P(r_i^k) = P(r_i^1 | i \in M_1) P(i \in M_1) + P(r_i^2 | i \in M_2) P(i \in M_2)$$

E-step



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

Probability that a data point (x_i, y_i) belongs to model k:

$$P(i \in M_k | r_i^k) = \frac{P(r_i^k | i \in M_k) P(i \in M_k)}{P(r_i^k)}$$

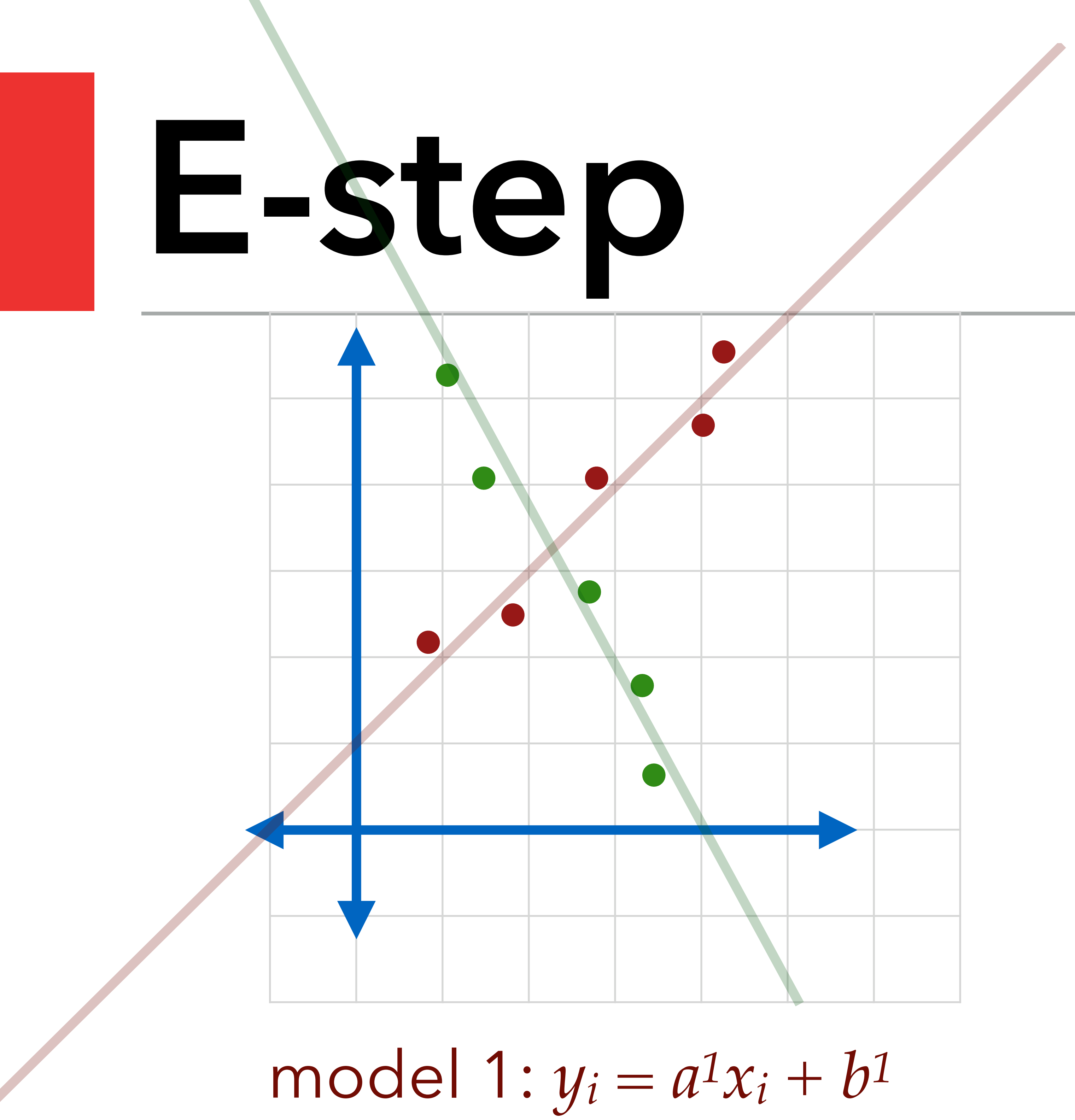
According to the law of total probability:

$$P(r_i^k) = P(r_i^1 | i \in M_1) P(i \in M_1) + P(r_i^2 | i \in M_2) P(i \in M_2)$$

Let's assume that model probabilities are equal, so:

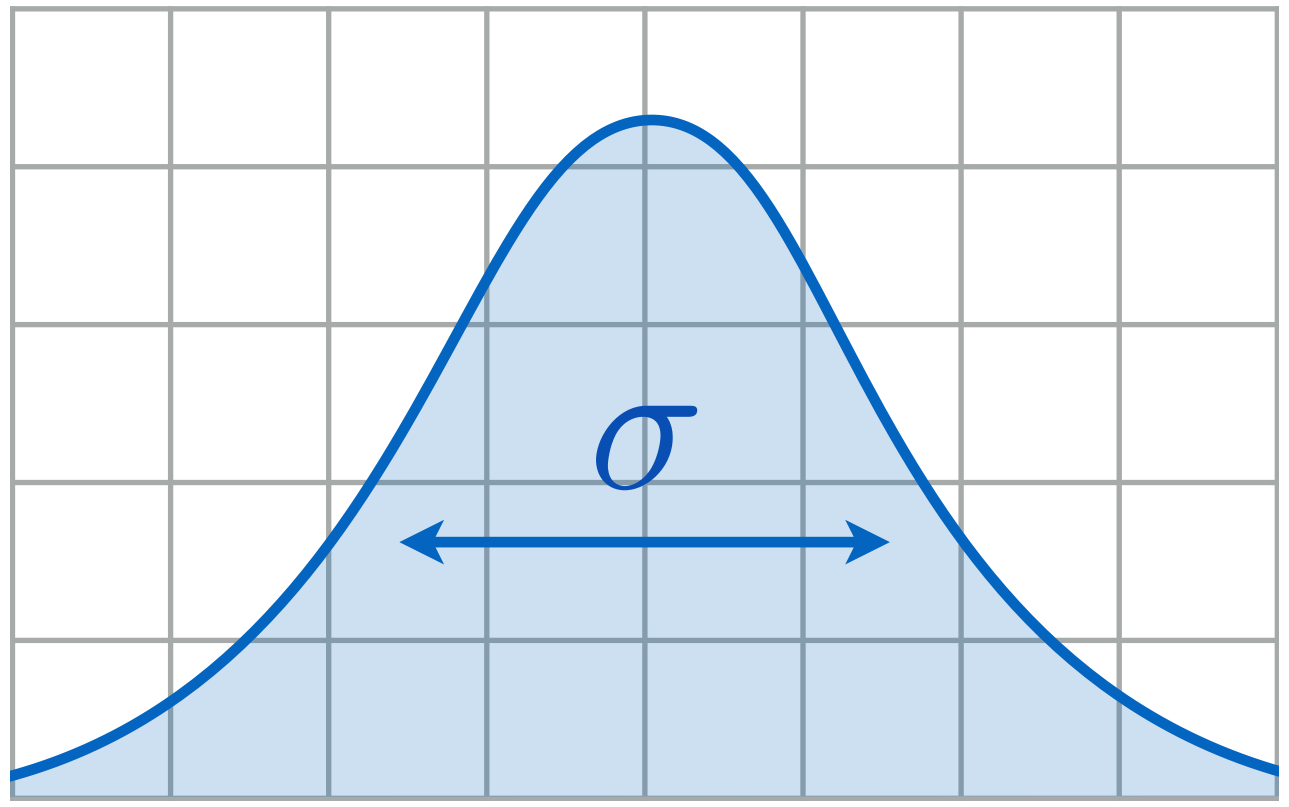
$$P(i \in M_k | r_i^k) = \frac{P(r_i^k | i \in M_k)}{P(r_i^1 | i \in M_1) + P(r_i^2 | i \in M_2)}$$

E-step



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$



0-mean normal (Gaussian) distribution

E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

Probability that a data point (x_i, y_i) belongs to model k:

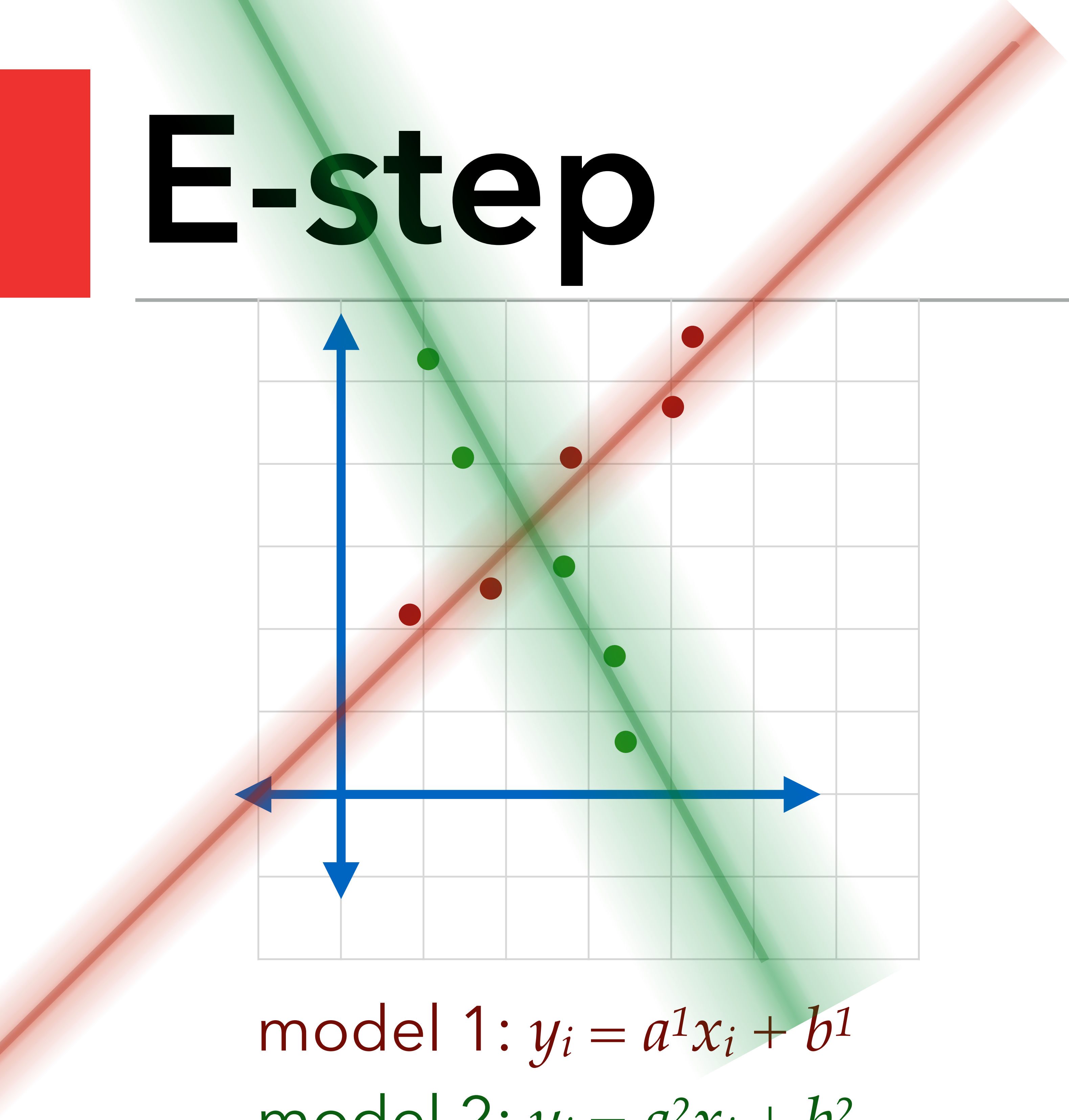
$$P(i \in M_k | r_i^k) = \frac{P(r_i^k | i \in M_k)}{P(r_i^1 | i \in M_1) + P(r_i^2 | i \in M_2)}$$

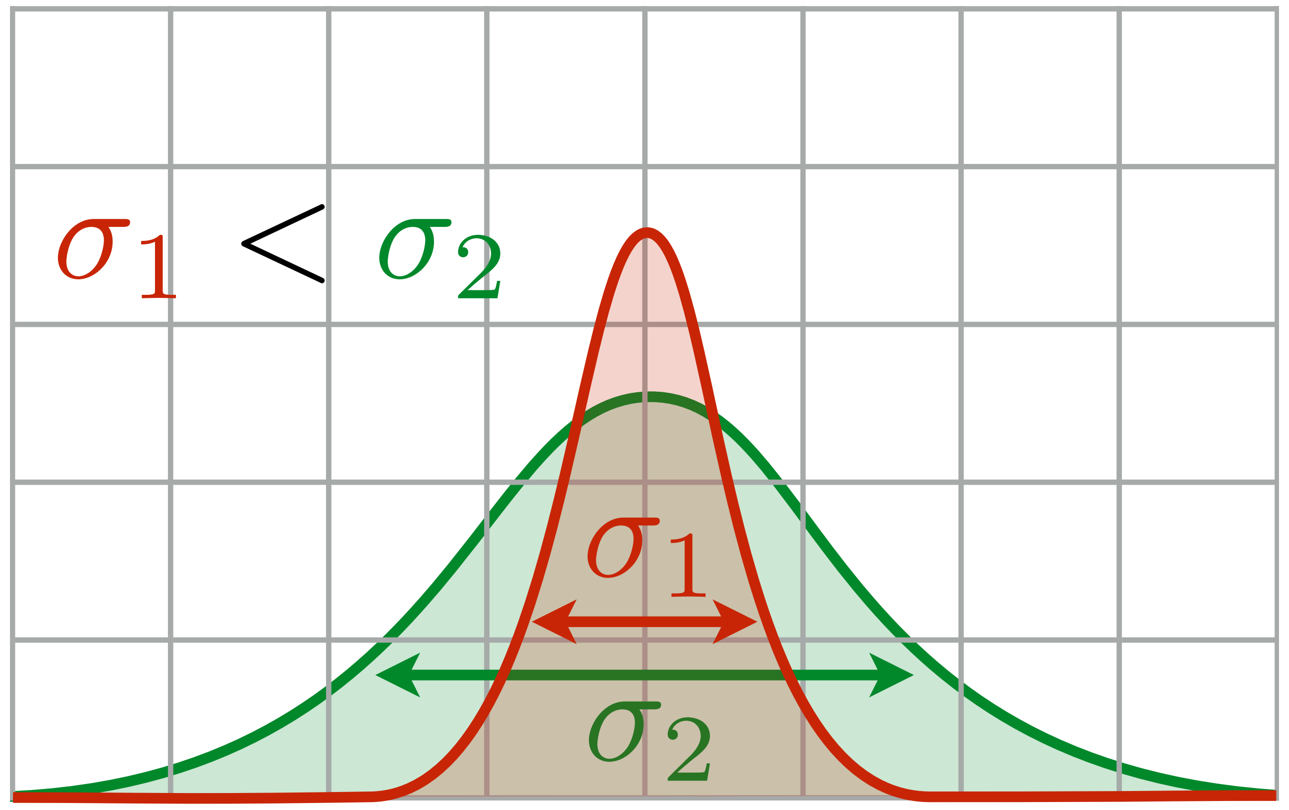
What is $P(r_i^k | i \in M_k)$?

… or, how much noise is in the data?

$$P(r_i^k | i \in M_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(r_i^k)^2 / 2\sigma_k^2}$$

E-step





0-mean normal (Gaussian) distribution

E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

Probability that a data point (x_i, y_i) belongs to model k:

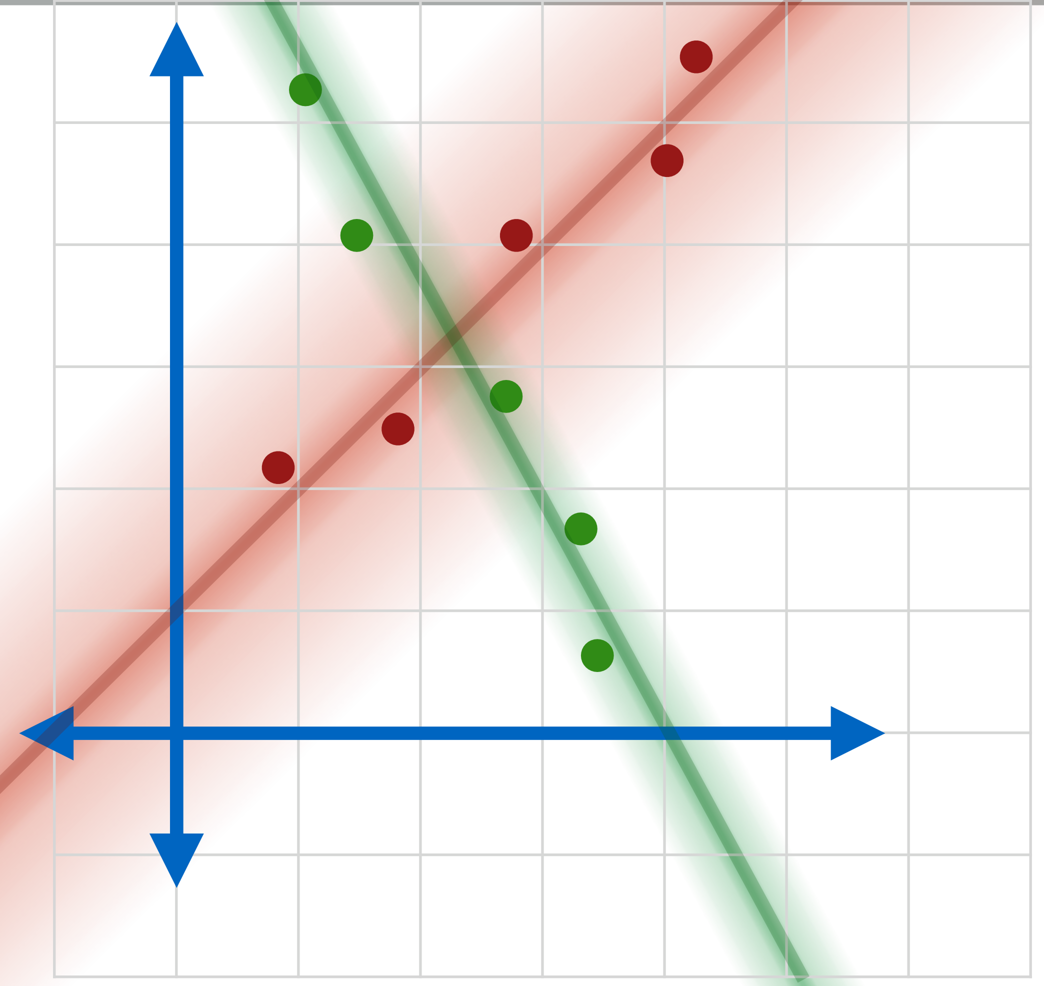
$$P(i \in M_k | r_i^k) = \frac{P(r_i^k | i \in M_k)}{P(r_i^1 | i \in M_1) + P(r_i^2 | i \in M_2)}$$

What is $P(r_i^k | i \in M_k)$?

… or, how much noise is in the data?

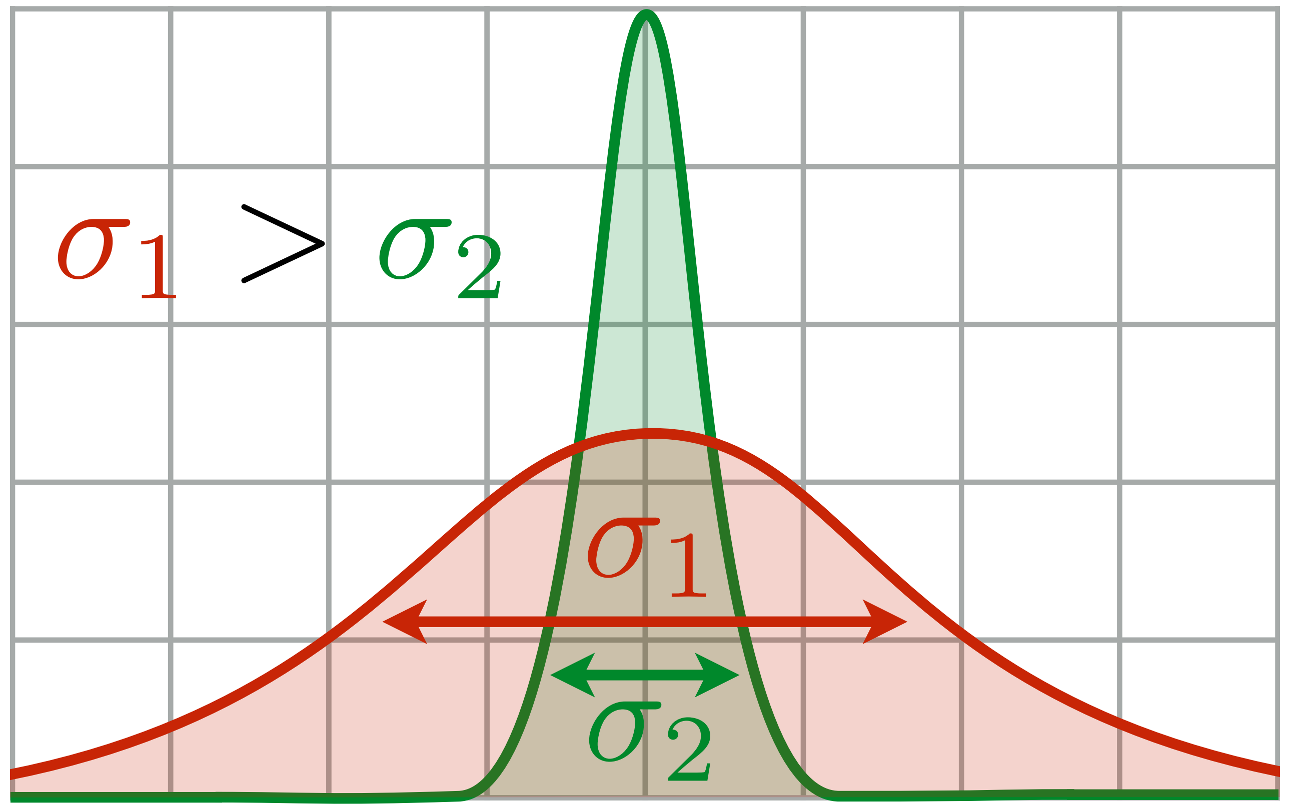
$$P(r_i^k | i \in M_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(r_i^k)^2 / 2\sigma_k^2}$$

E-step



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$



0-mean normal (Gaussian) distribution

E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

Probability that a data point (x_i, y_i) belongs to model k:

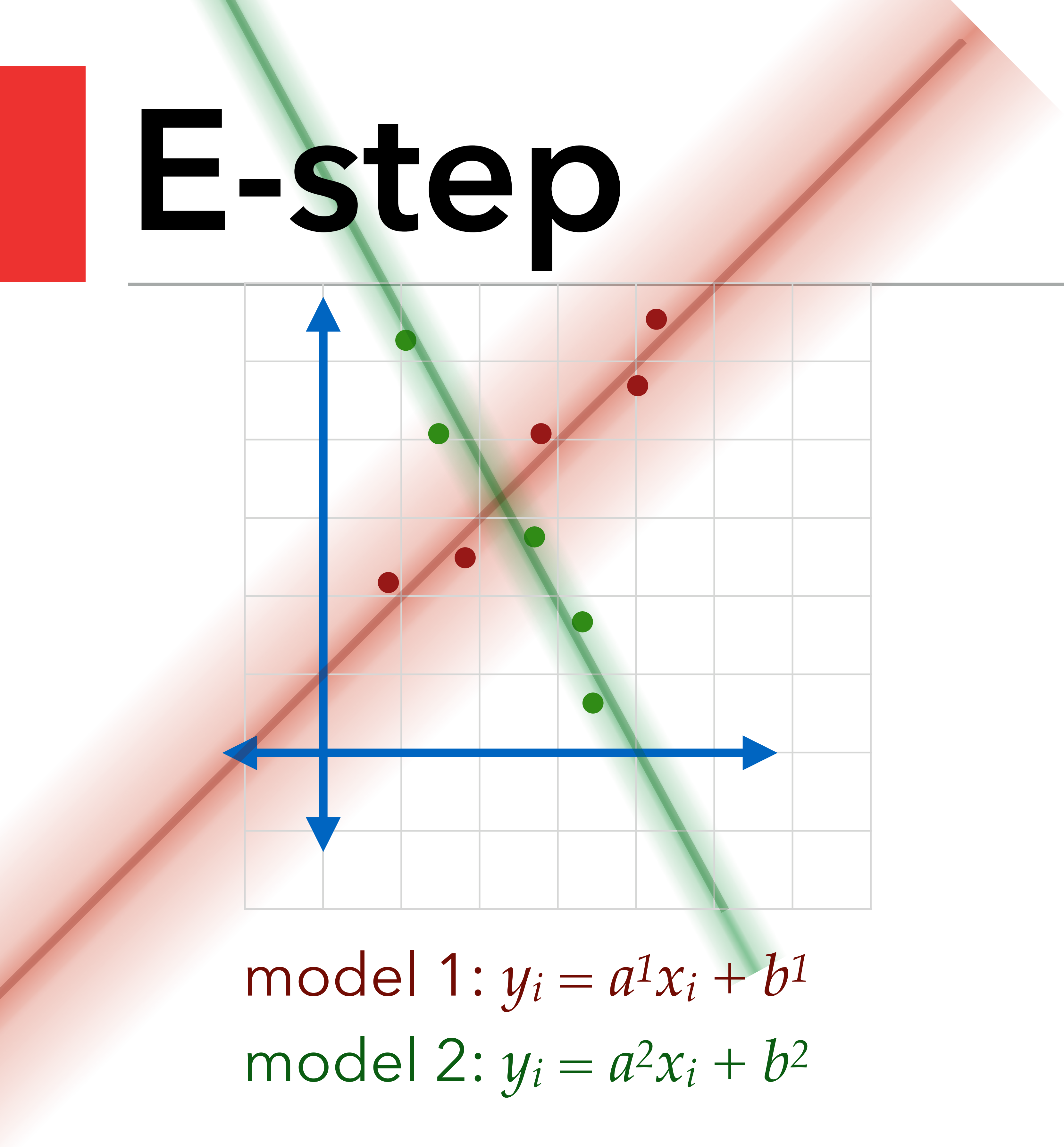
$$P(i \in M_k | r_i^k) = \frac{P(r_i^k | i \in M_k)}{P(r_i^1 | i \in M_1) + P(r_i^2 | i \in M_2)}$$

What is $P(r_i^k | i \in M_k)$?

… or, how much noise is in the data?

$$P(r_i^k | i \in M_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(r_i^k)^2 / 2\sigma_k^2}$$

E-step



E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

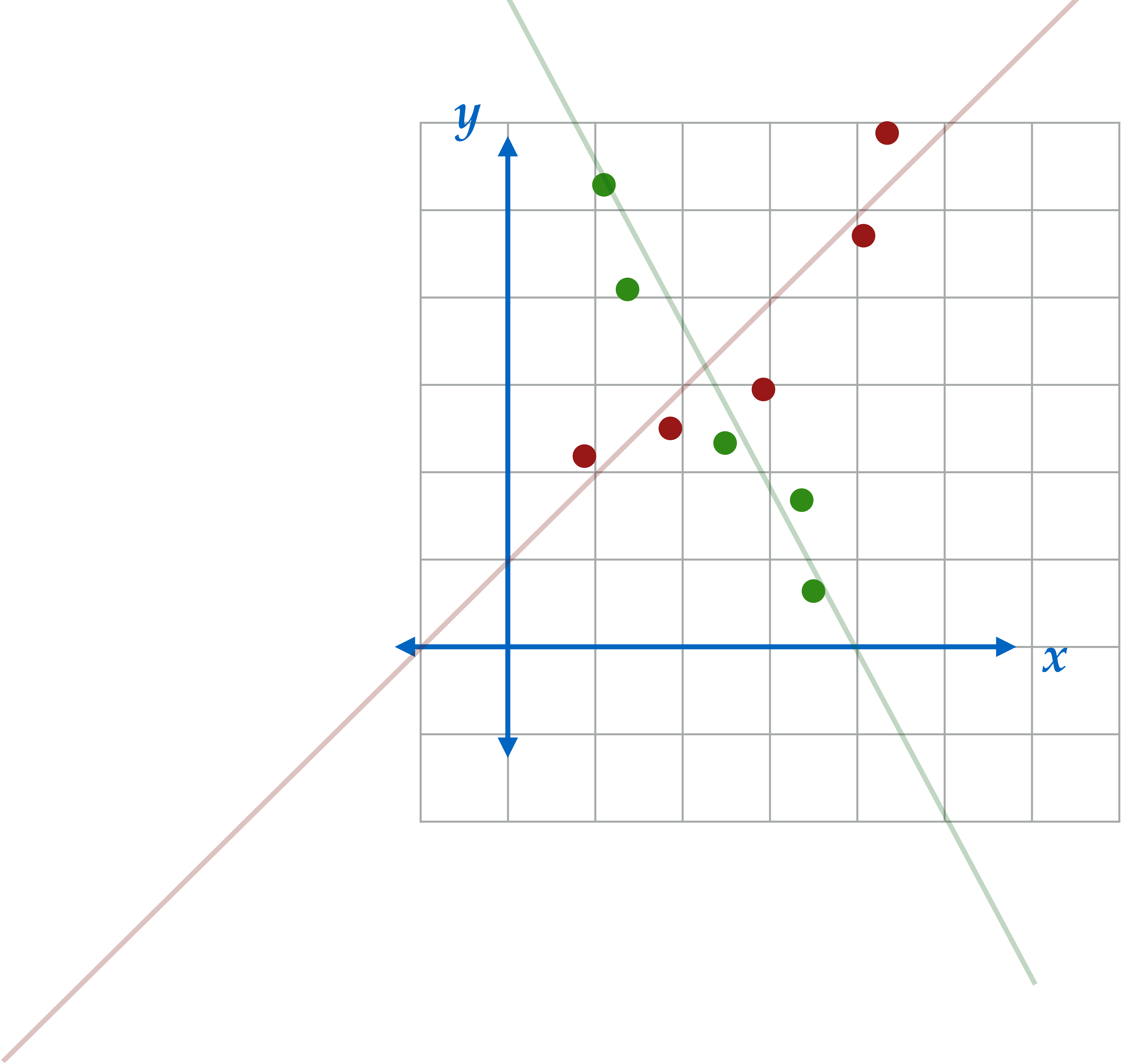
Compute the residual for each data point i, and for each model k, e.g., $r_i^k = |a^k x_i + b^k - y_i|$

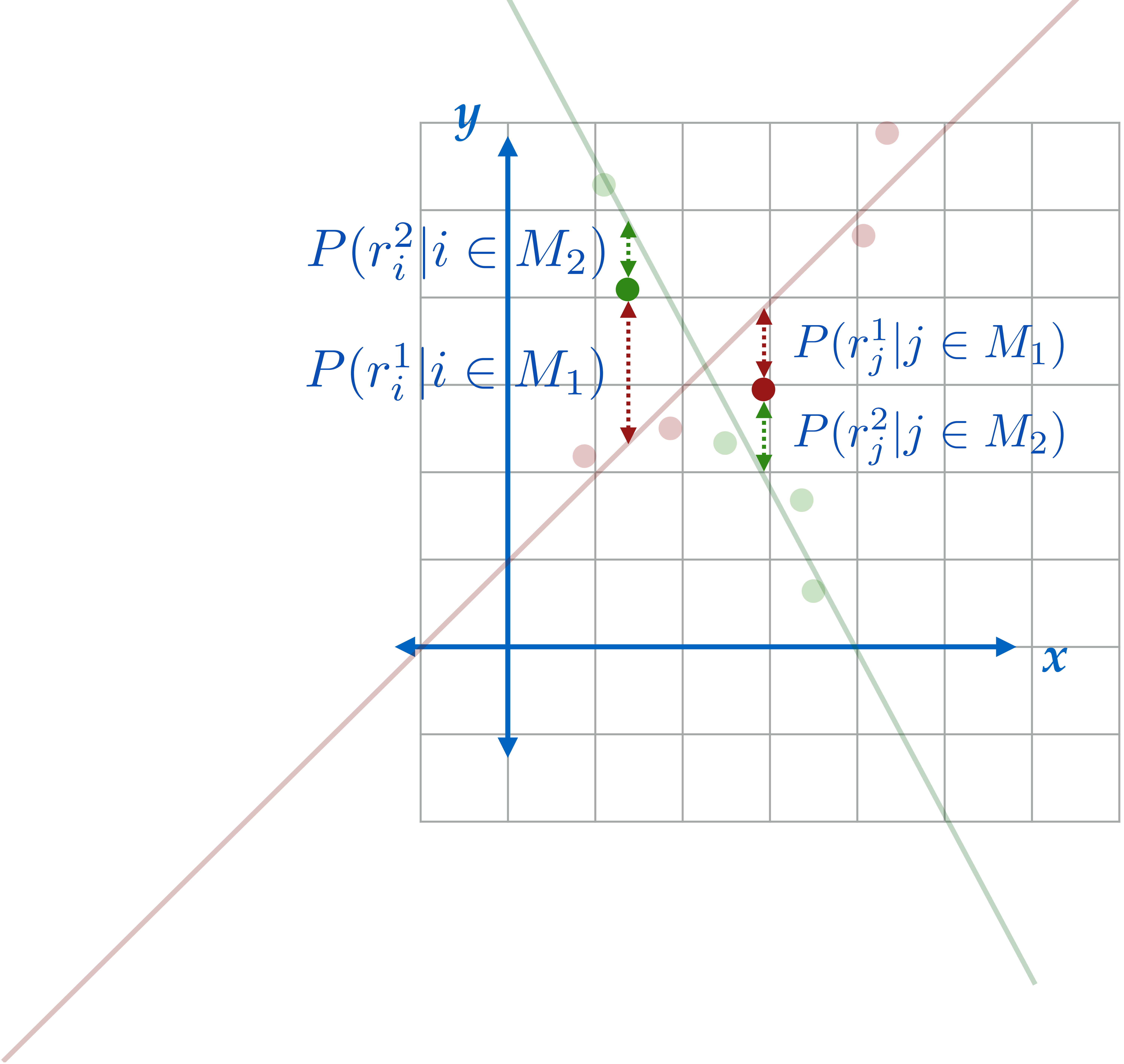
Compute the probability of each data point i belonging to each model k:

$$P(i \in M_k | r_i^k) = \frac{P(r_i^k | i \in M_k)}{P(r_i^1 | i \in M_1) + P(r_i^2 | i \in M_2)}$$

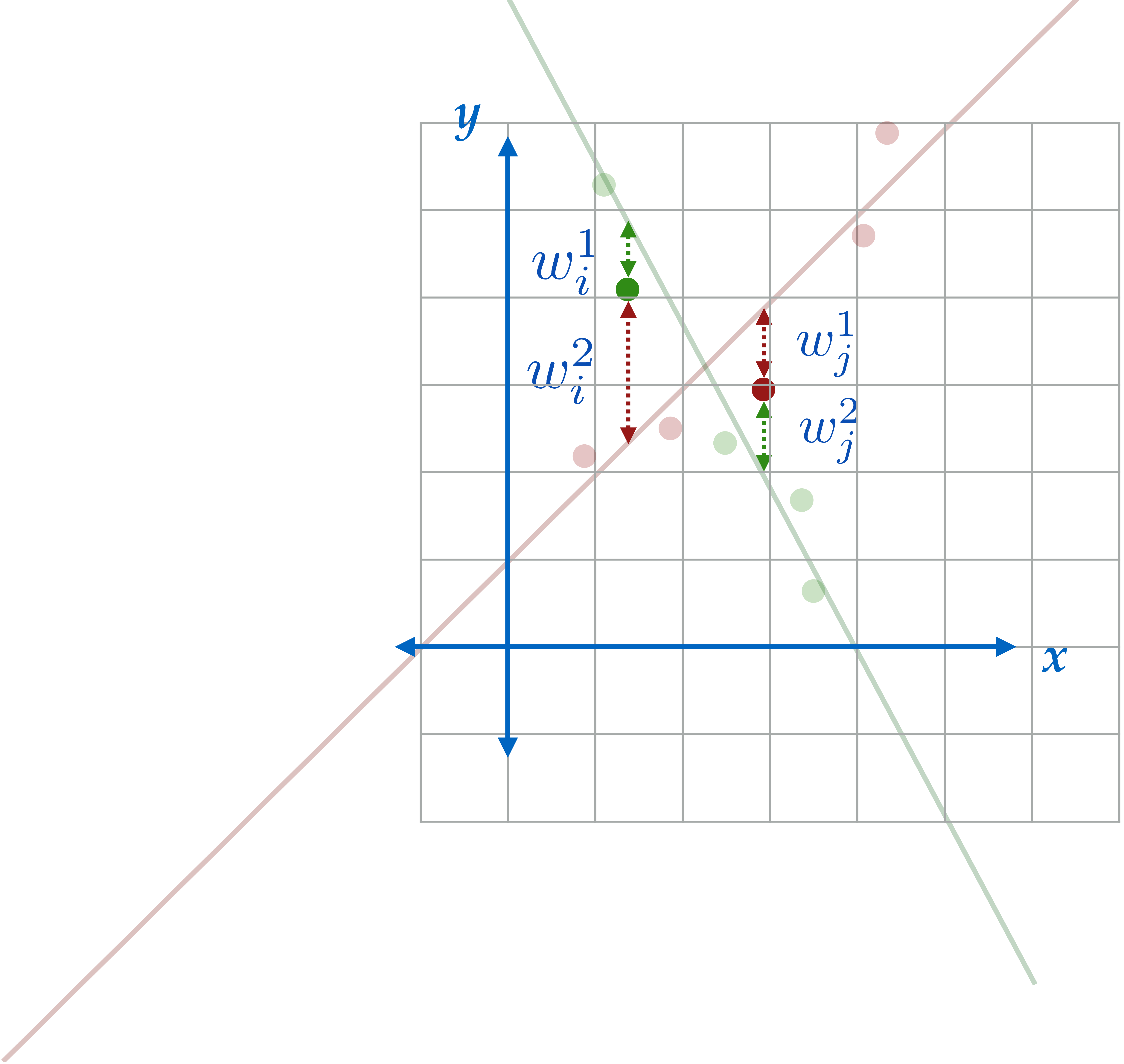
with, for example, $P(r_i^k | i \in M_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(r_i^k)^2/2\sigma_k^2}$

Note: the probability of belonging to model 1 or 2 sums to 1: $a/(a+b) + b/(a+b) = 1$


y
x

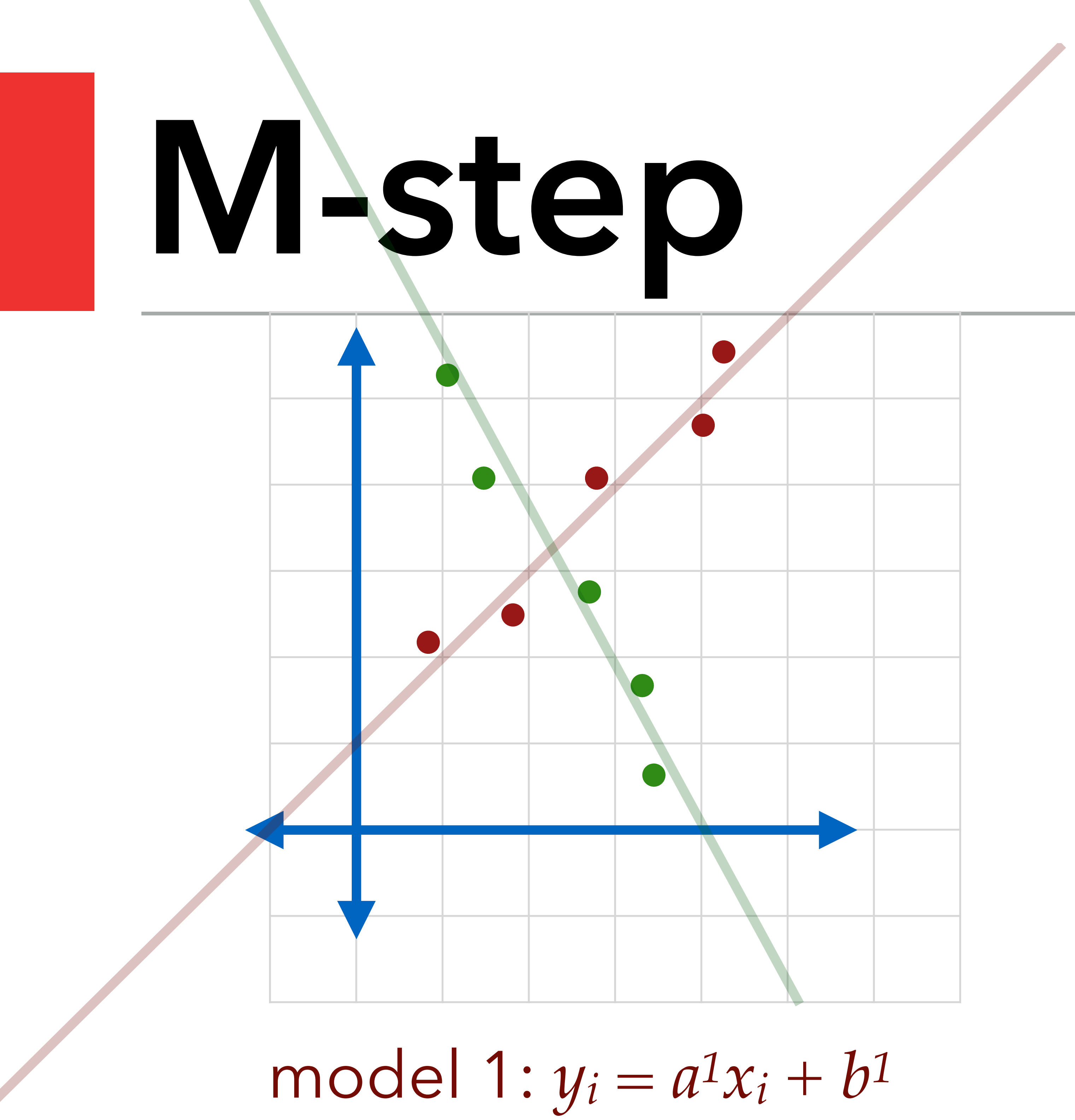


y
$P(r_i^2|i \in M_2)$
$P(r_i^1|i \in M_1)$
$P(r_j^1|j \in M_1)$
$P(r_j^2|j \in M_2)$
x



y
x
w_i^1
w_i^2
w_j^1
w_j^2

M-step



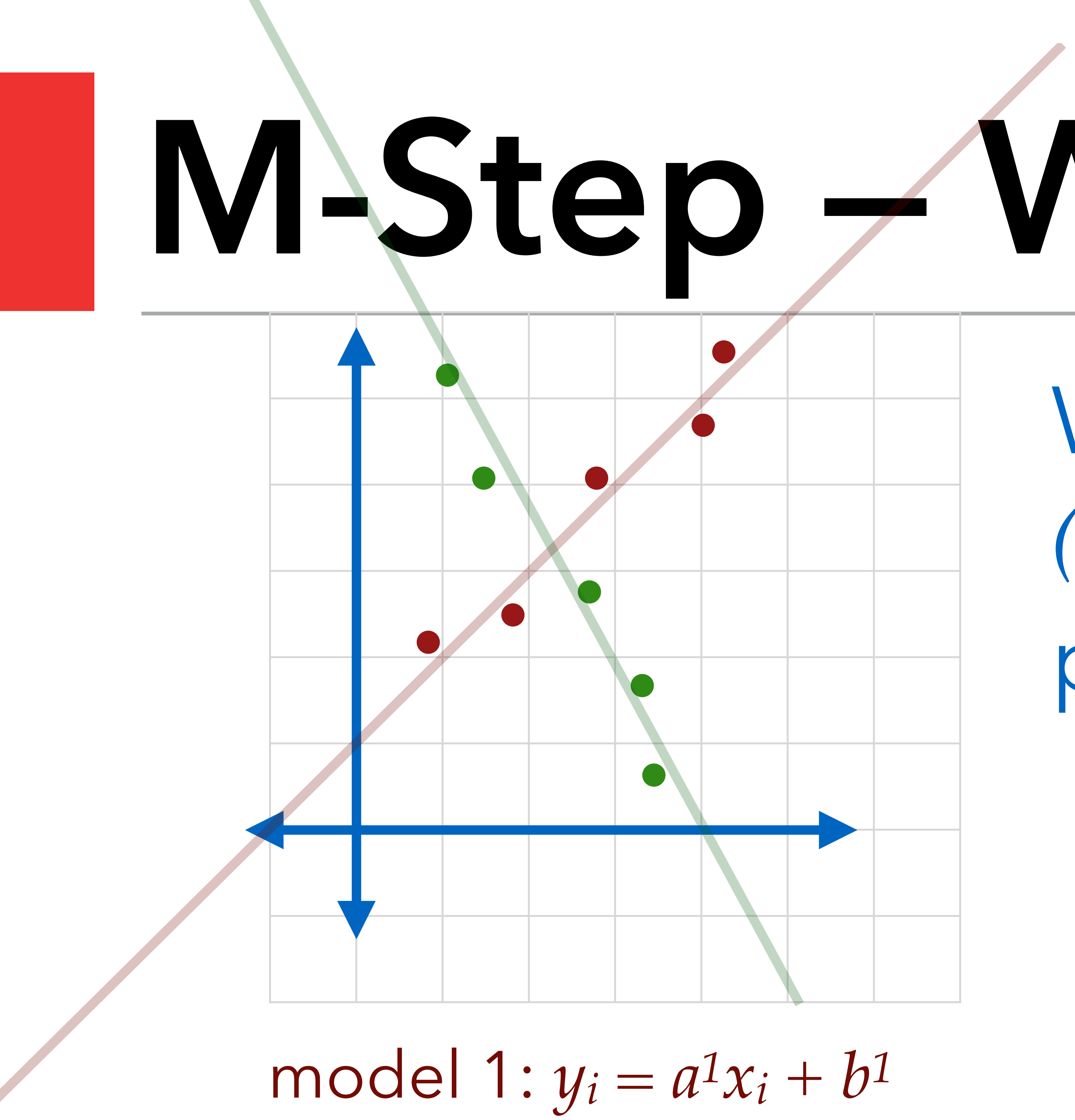
model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

M-step: re-estimate model parameters for each model (e.g., $k = 1, 2$) using probabilistic assignments

M-Step – WLS Solution Formulation

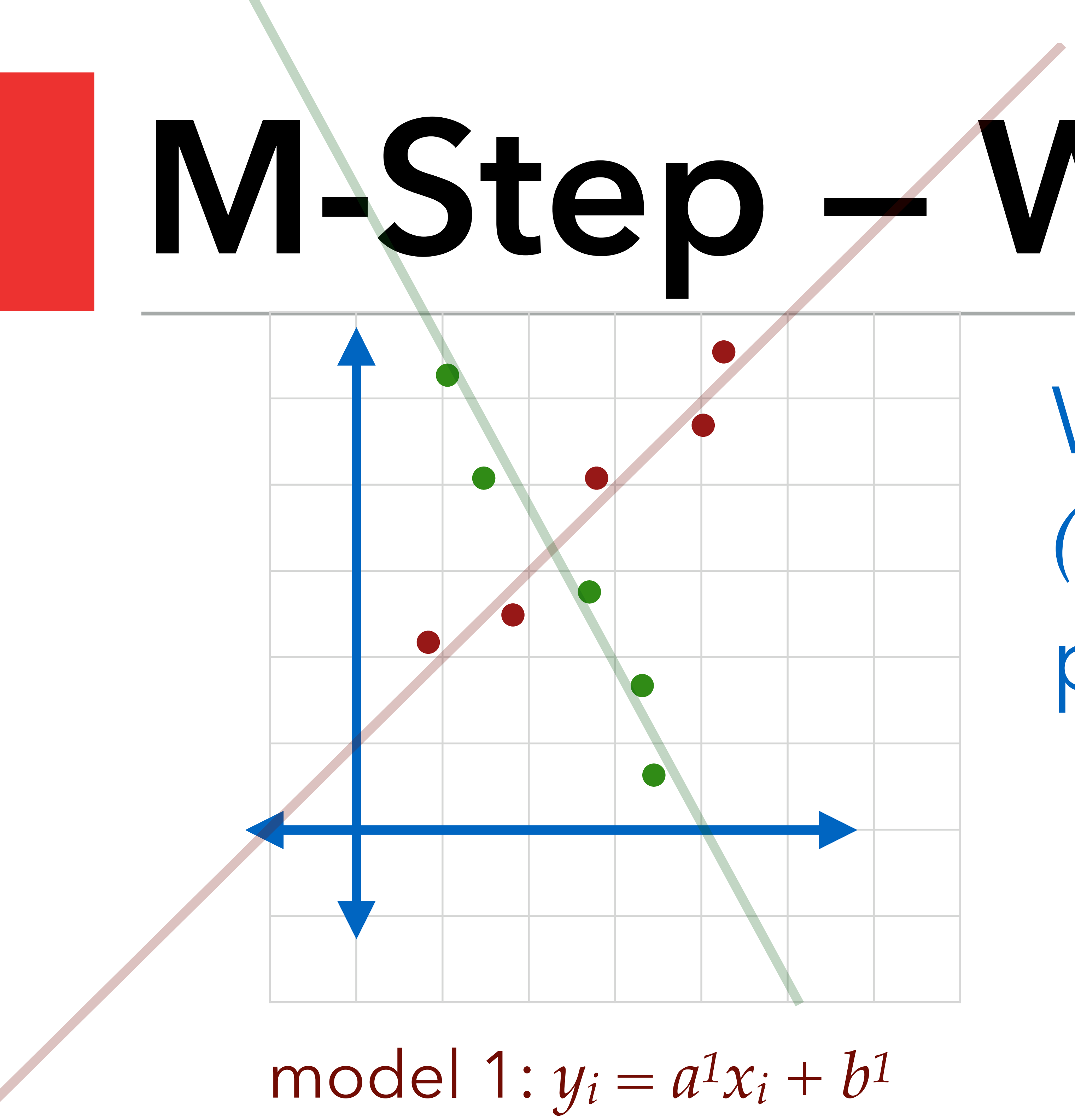


model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

Write a weighted least-squares error function to estimate (a^k, b^k), where the "weight" of each point (x_i, y_i) is the probability computed in the **E-step**, w_i^k

M-Step – WLS Solution Formulation



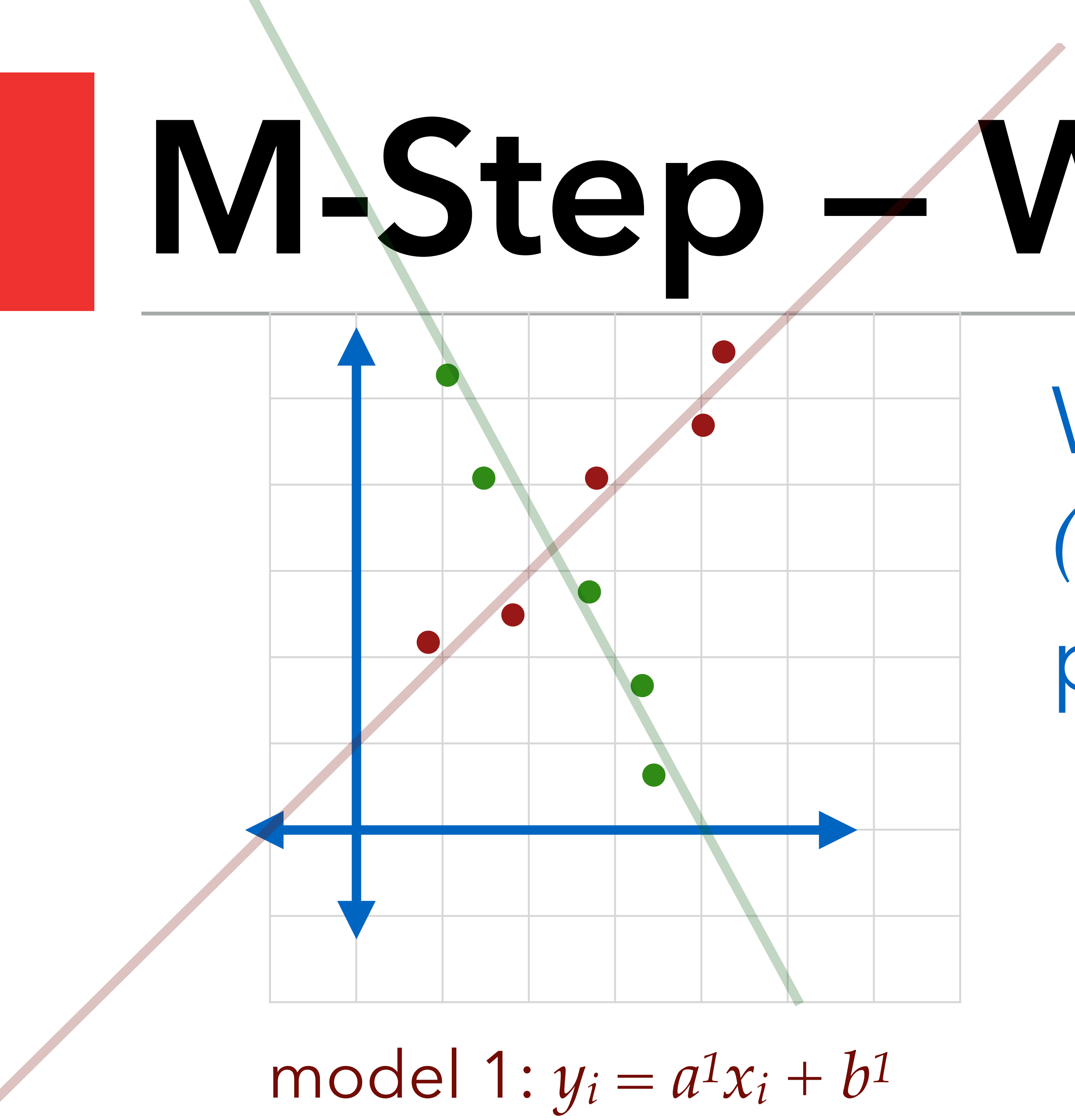
model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

Write a weighted least-squares error function to estimate (a^k, b^k), where the "weight" of each point (x_i, y_i) is the probability computed in the **E-step**, w_i^k

$$E(a^k, b^k) = \sum_{i=1}^{n} (w_i^k (a^k x_i + b^k - y_i))^2$$

M-Step – WLS Solution Formulation



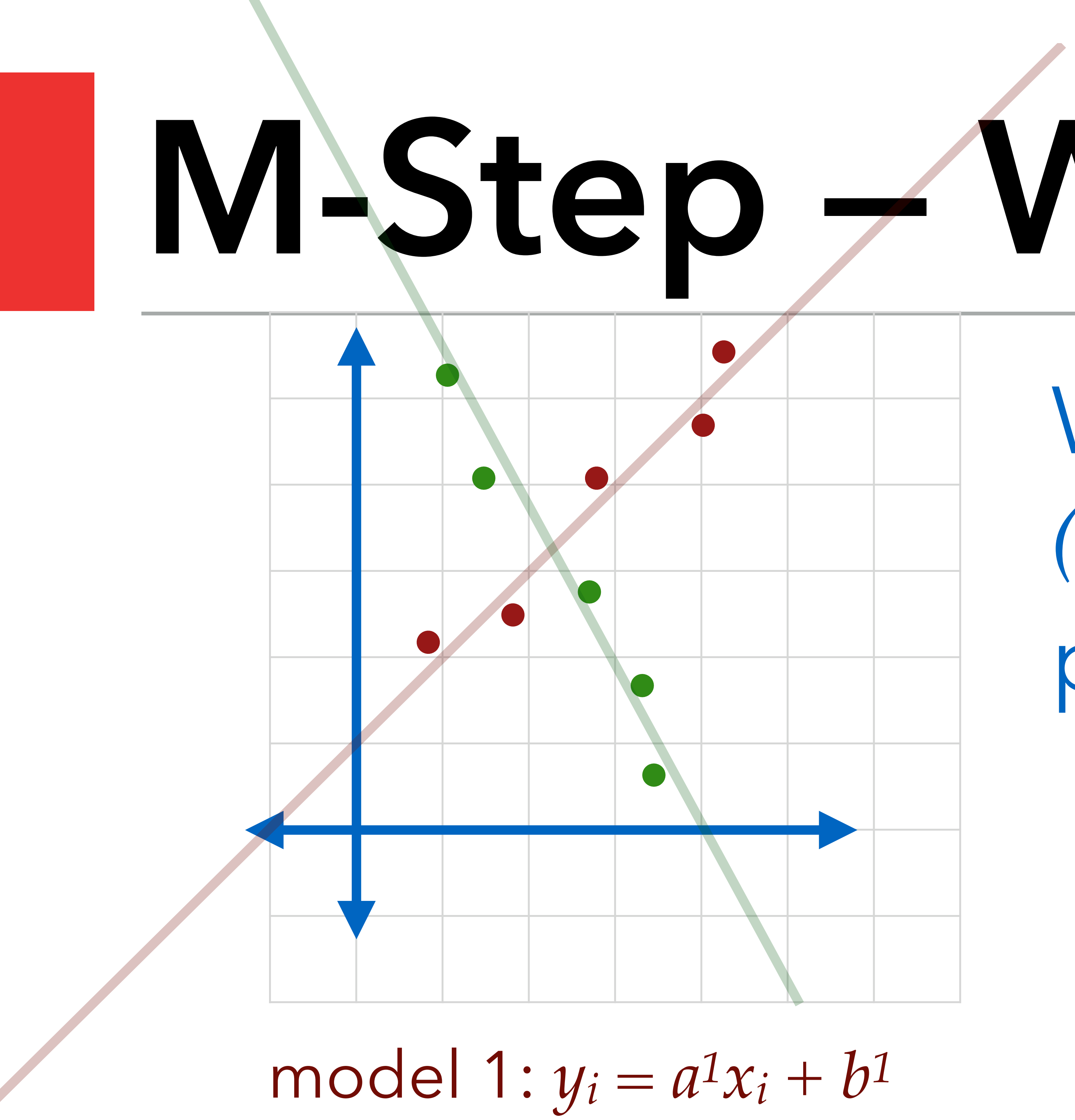
model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

Write a weighted least-squares error function to estimate (a^k, b^k), where the “weight” of each point (x_i, y_i) is the probability computed in the **E-step**, w_i^k

$$E(a^k, b^k) = \sum_{i=1}^{n} (w_i^k (a^k x_i + b^k - y_i))^2$$

M-Step – WLS Solution Formulation



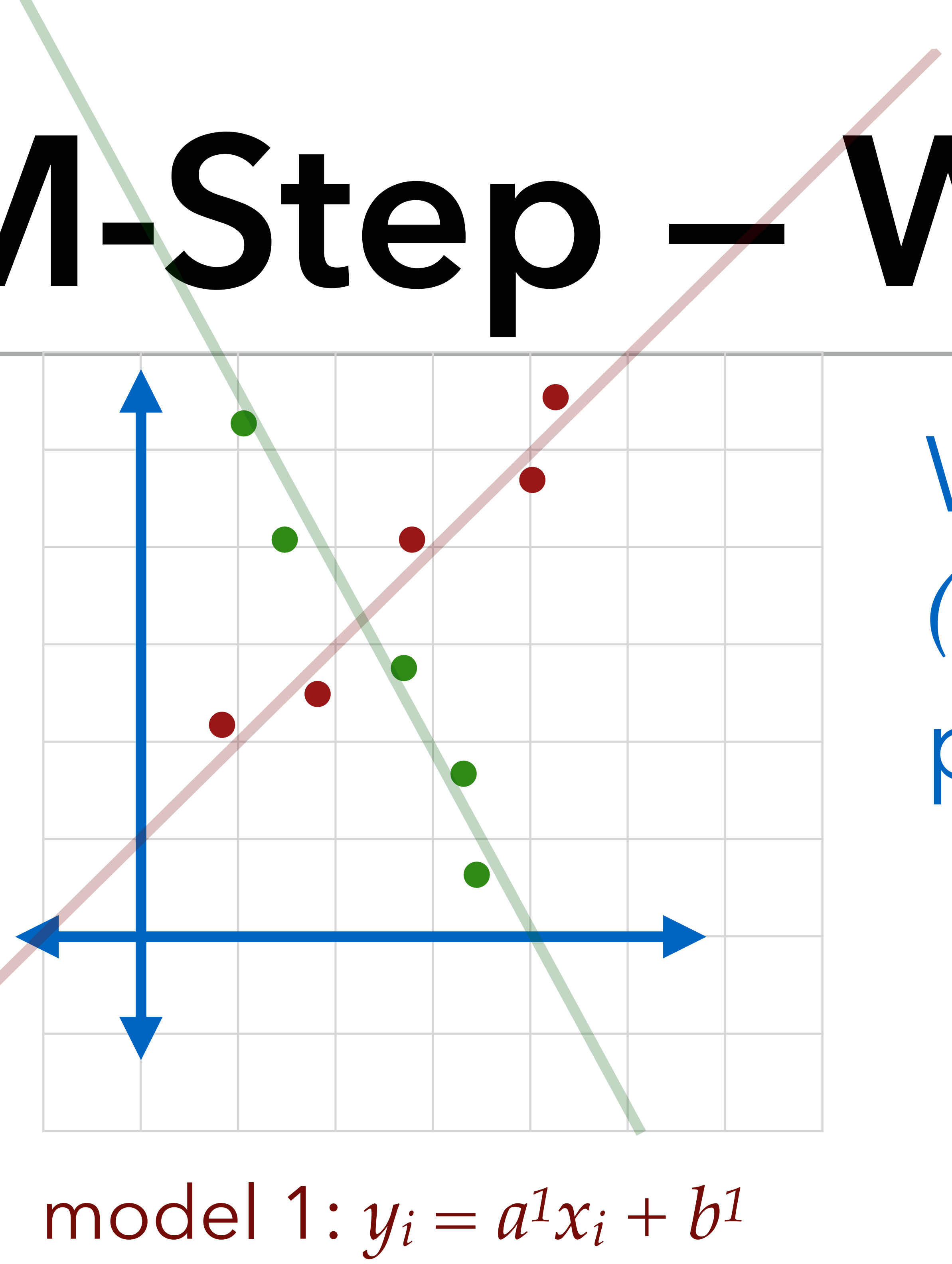
model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

Write a weighted least-squares error function to estimate (a^k, b^k), where the "weight" of each point (x_i, y_i) is the probability computed in the **E-step**, w_i^k

$$E(a^k, b^k) = \sum_{i=1}^{n} (w_i^k (a^k x_i + b^k - y_i))^2$$

M-Step – WLS Solution Formulation



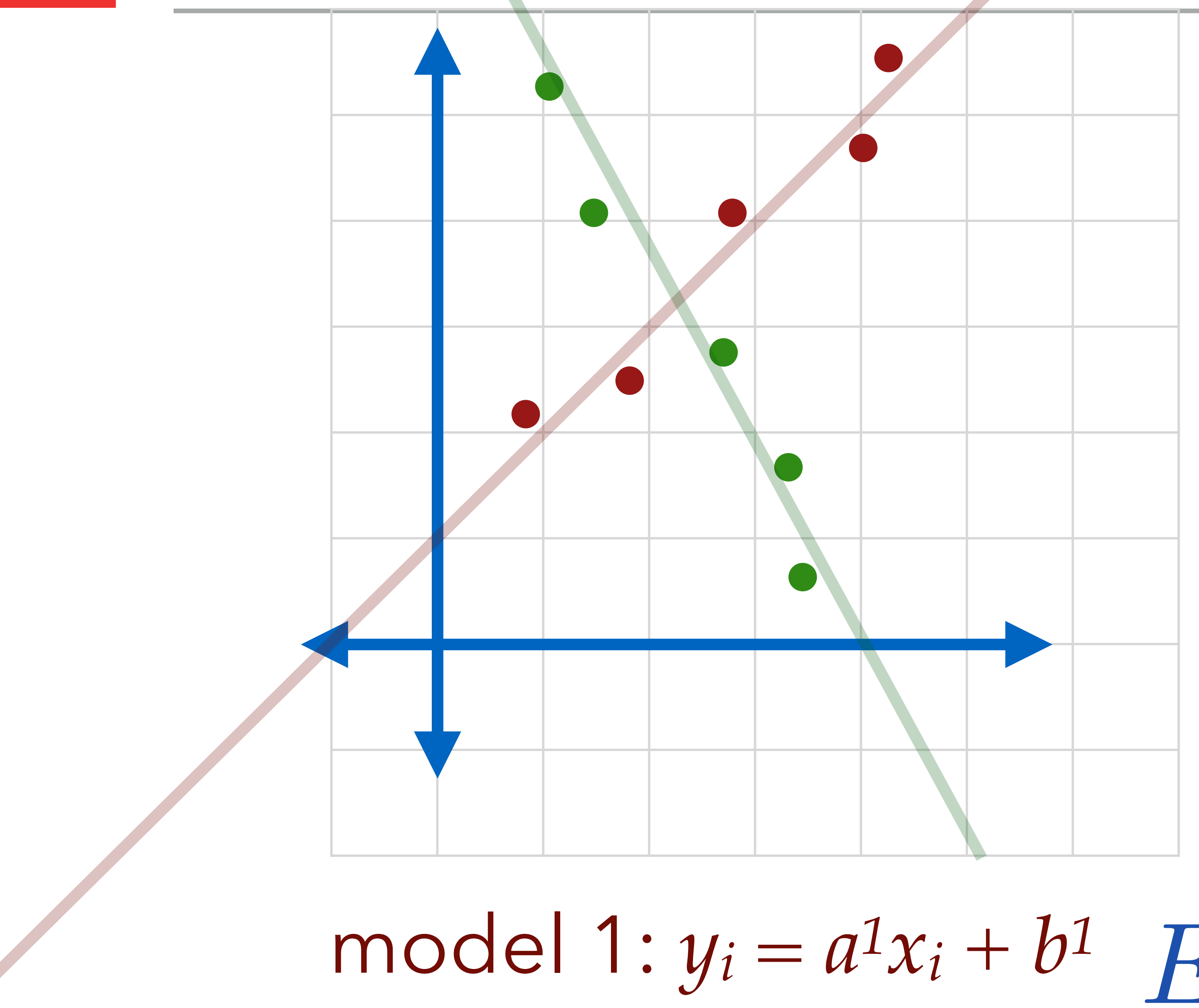
model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

Write a weighted least-squares error function to estimate (a^k, b^k), where the "weight" of each point (x_i, y_i) is the probability computed in the **E-step**, w_i^k

$$E(a^k, b^k) = \sum_{i=1}^{n} \left(w_i^k (a^k x_i + b^k - y_i) \right)^2$$

M-Step – WLS Solution Formulation



model 1: $y_i = a^1 x_i + b^1$

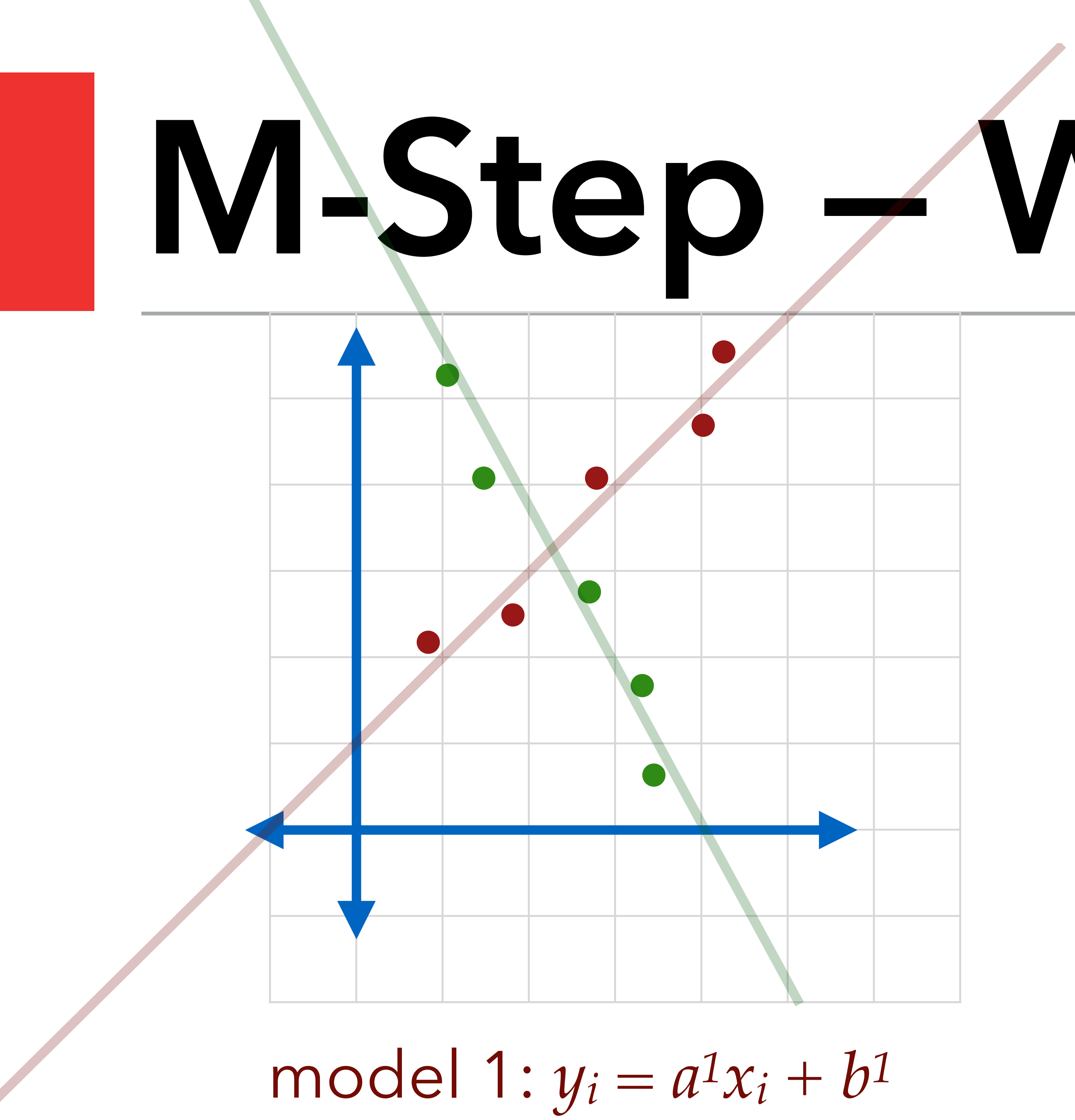
model 2: $y_i = a^2 x_i + b^2$

Express the weighted least-squares (WLS) error function in matrix form, differentiate with respect to the unknowns, set equal to zero, and solve for the WLS solution

$$E(\vec{m}^k) = \left\| \begin{pmatrix} w_1^k & 0 & \cdots & 0 \\ 0 & w_2^k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n^k \end{pmatrix} \left[\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} a^k \\ b^k \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \right] \right\|^2$$

$$E(\vec{m}^k) = \|W^k (X\vec{m}^k - \vec{y})\|^2$$

M-Step – WLS Solution Formulation



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

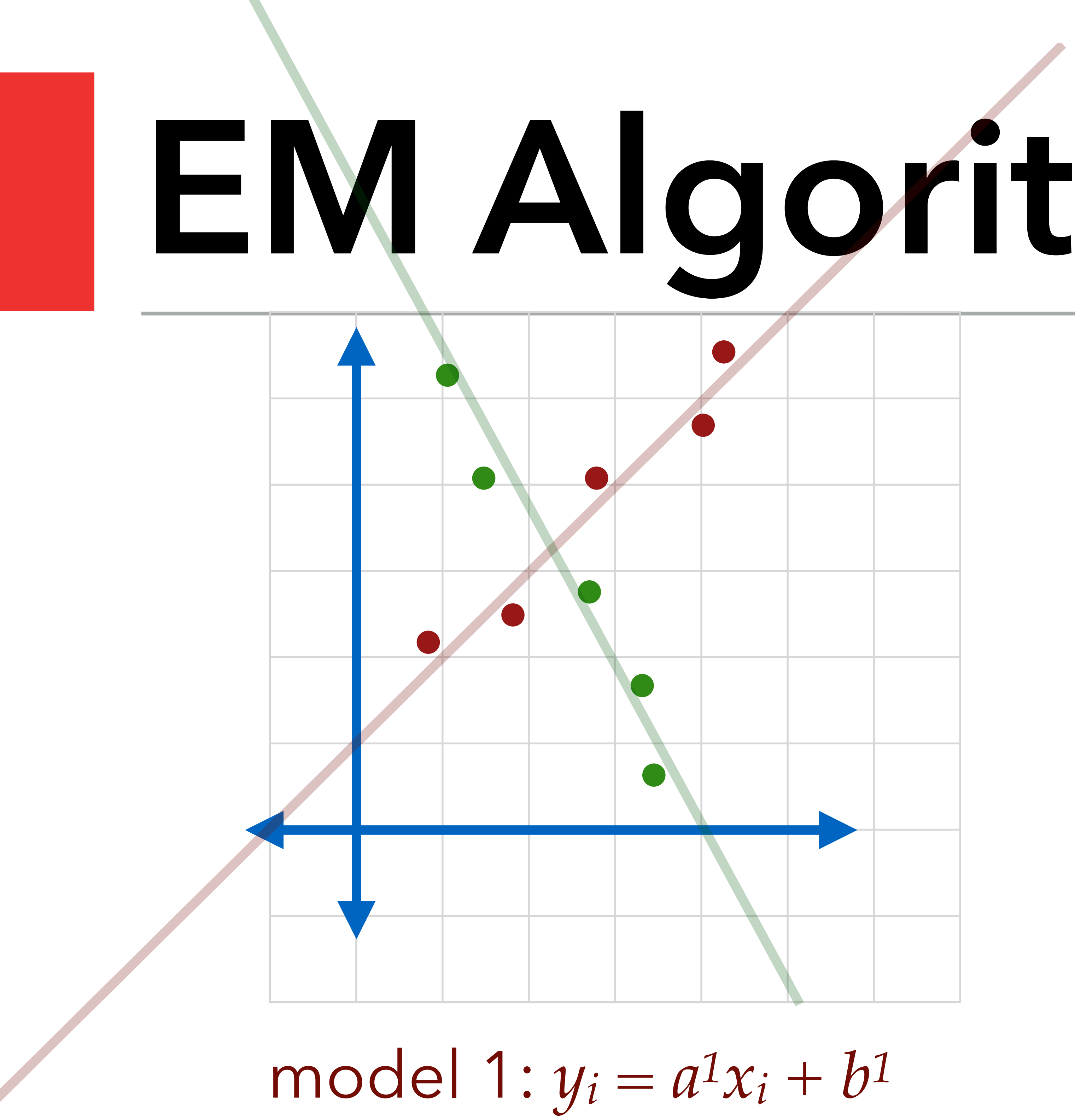
$$E(\vec{m}^k) = \|W^k(X\vec{m}^k - \vec{y})\|^2$$

$$\frac{dE(\vec{m}^k)}{d\vec{m}^k} = 2X^T(W_k)^T(W_k X\vec{m}^k - W_k\vec{y})$$

$$2(X^T W_k^2 X)\vec{m}^k - 2X^T W_k^2 \vec{y} = 0$$

$$\vec{m}^k = (X^T W_k^2 X)^{-1} X^T W_k^2 \vec{y}$$

EM Algorithm



model 1: $y_i = a^1 x_i + b^1$

model 2: $y_i = a^2 x_i + b^2$

E-step: *assume* the model parameters are known, and compute the probability of each data point (x_i, y_i) belonging to each model (e.g., $k = 1, 2$)

M-step: re-estimate model parameters for each model (e.g., $k = 1, 2$) using probabilistic assignments, e.g., with a WLS solve:

$$\vec{m}^k = (X^T W_k^2 X)^{-1} X^T W_k^2 \vec{y}$$

Iterate: repeat until a convergence criterion is met, e.g.,

- model parameters do not change much, or
- maximum residual is below a threshold

EM – Probabilistic Interpretation

EM – Probabilistic Derivation

We can re-interpret EM as an MLE or MAP estimator, using a more probabilistic interpretation of each EM stage

- **E-step** estimates the expected (log-)likelihood of the data, given the (current) parameters
- **M-step** solves for parameters that maximize this (log-)likelihood
- recall how Bayes' allows us to relate the posterior and likelihood

Assignment 2 provides one such interpretation in the context of Gaussian mixture models

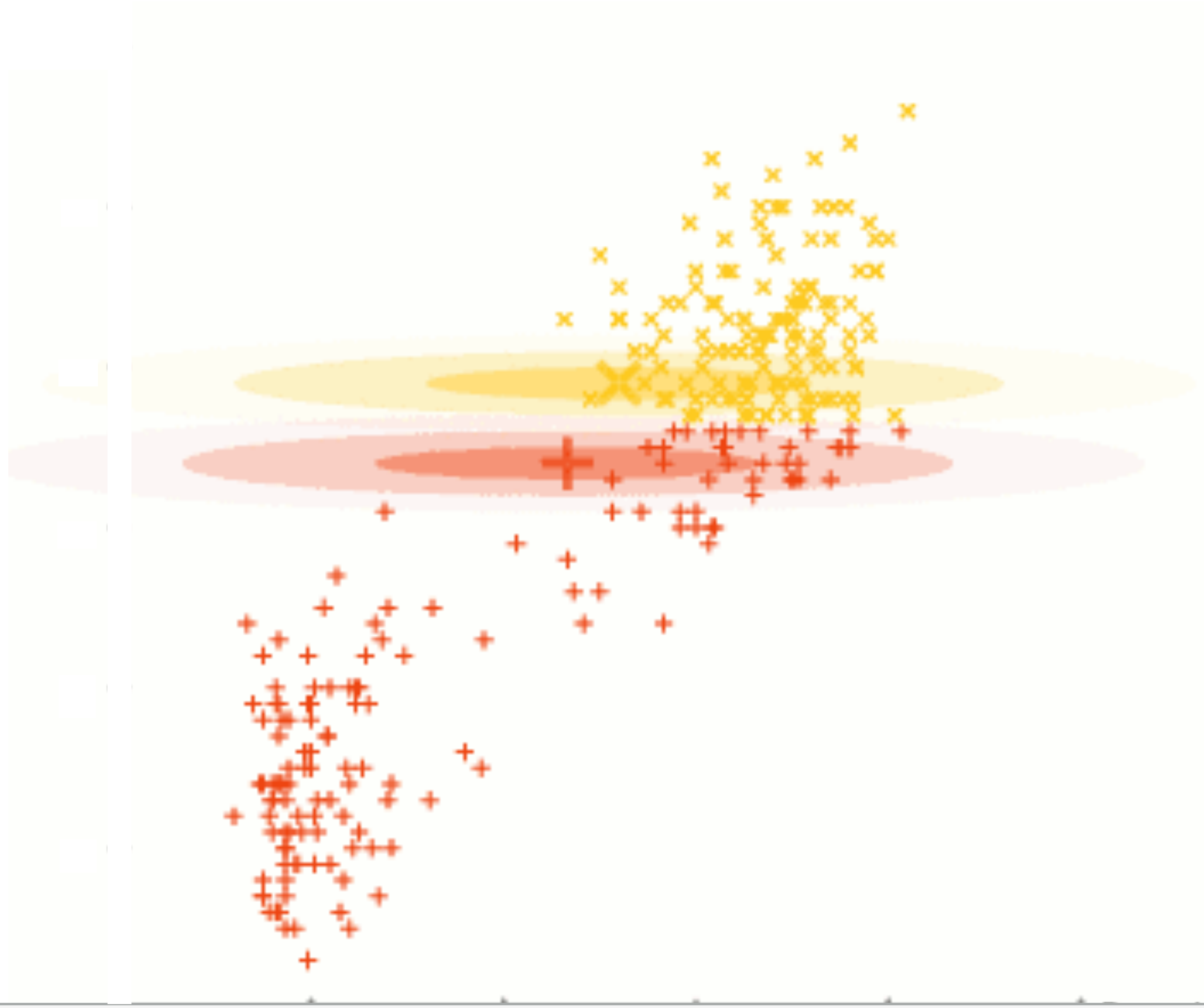
EM applied

EM Applications

EM is a very powerful and flexible tool

- 🙂 applies to **any** underlying, parameterized model
  - polynomials, multi-variate Gaussians, sinusoids, mixtures…
  - works as MLE/MAP when their direct derivation isn't possible
- 🙂 can yield good solutions with few iterations
  - may be slow to converge, but a few iterations may sufficient
- ☹️ need to know *how many* components your mixture model has
- ☹️ sensitive to initialization

EM Example




256 Gaussians