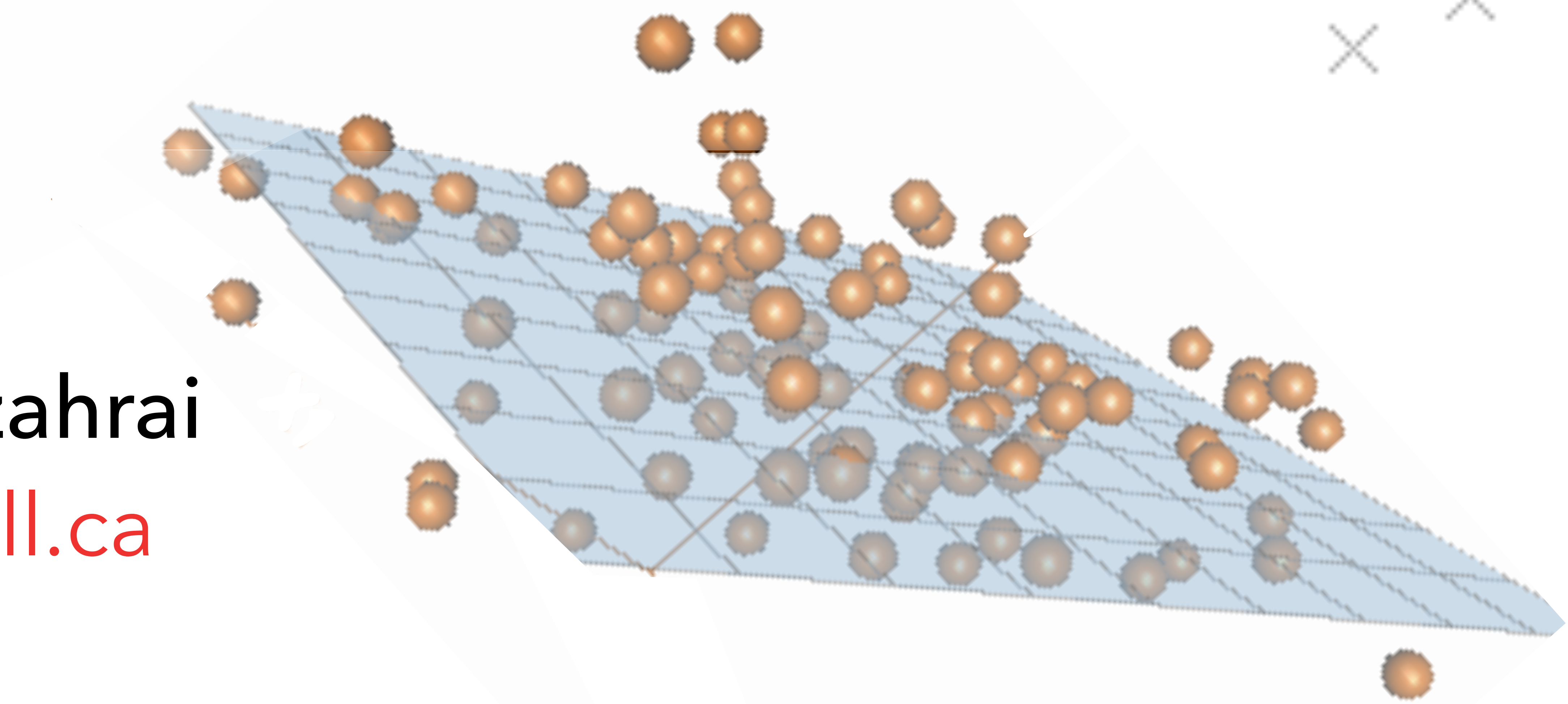
2B – LEAST-SQUARES: OVERDETERMINED SYSTEMS

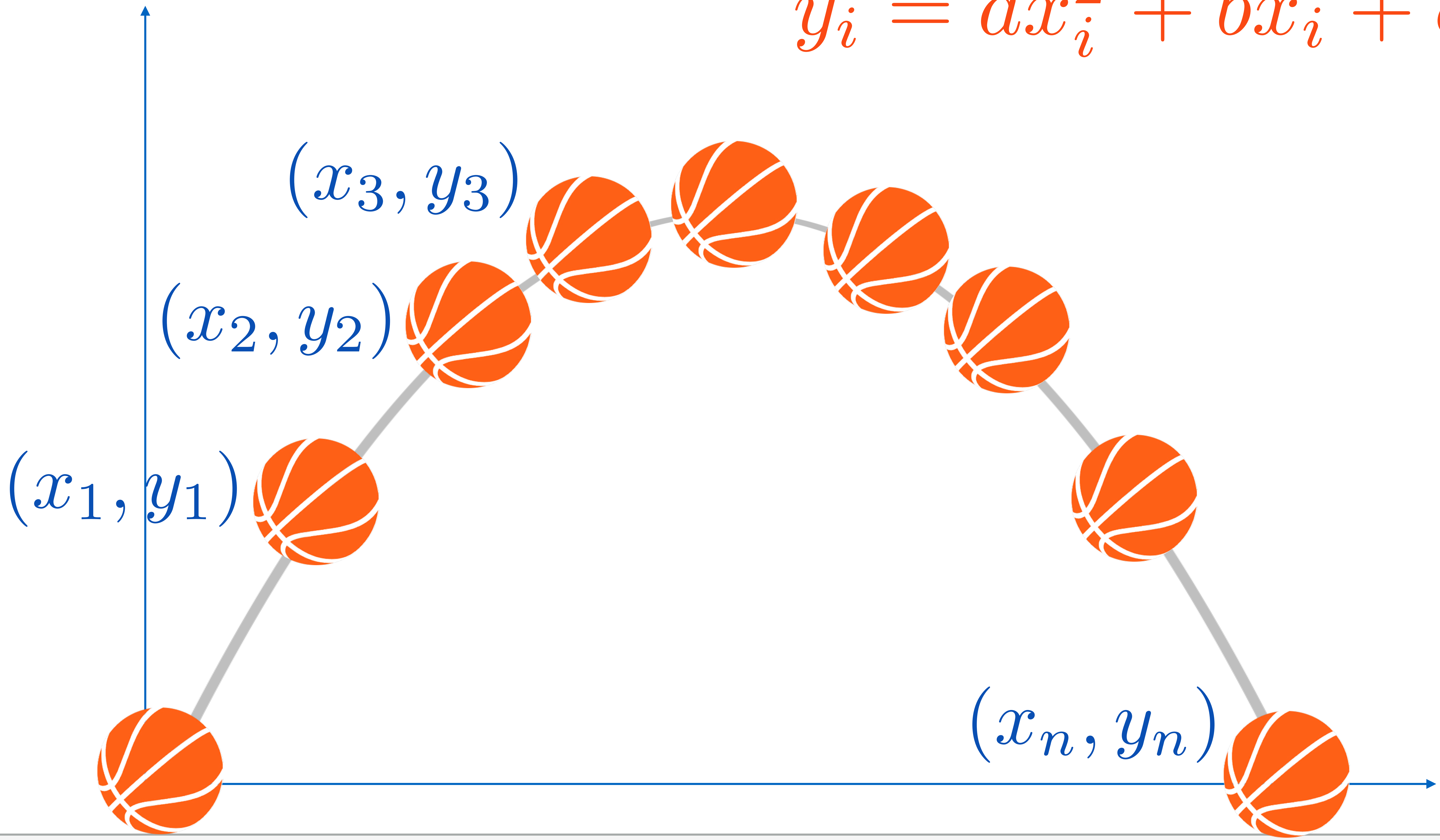
Derek Nowrouzezahrai
derek@cim.mcgill.ca



Overdetermined Linear Systems & Least-squares

Recall – sampled ballistic motion

$$y_i = ax_i^2 + bx_i + c$$



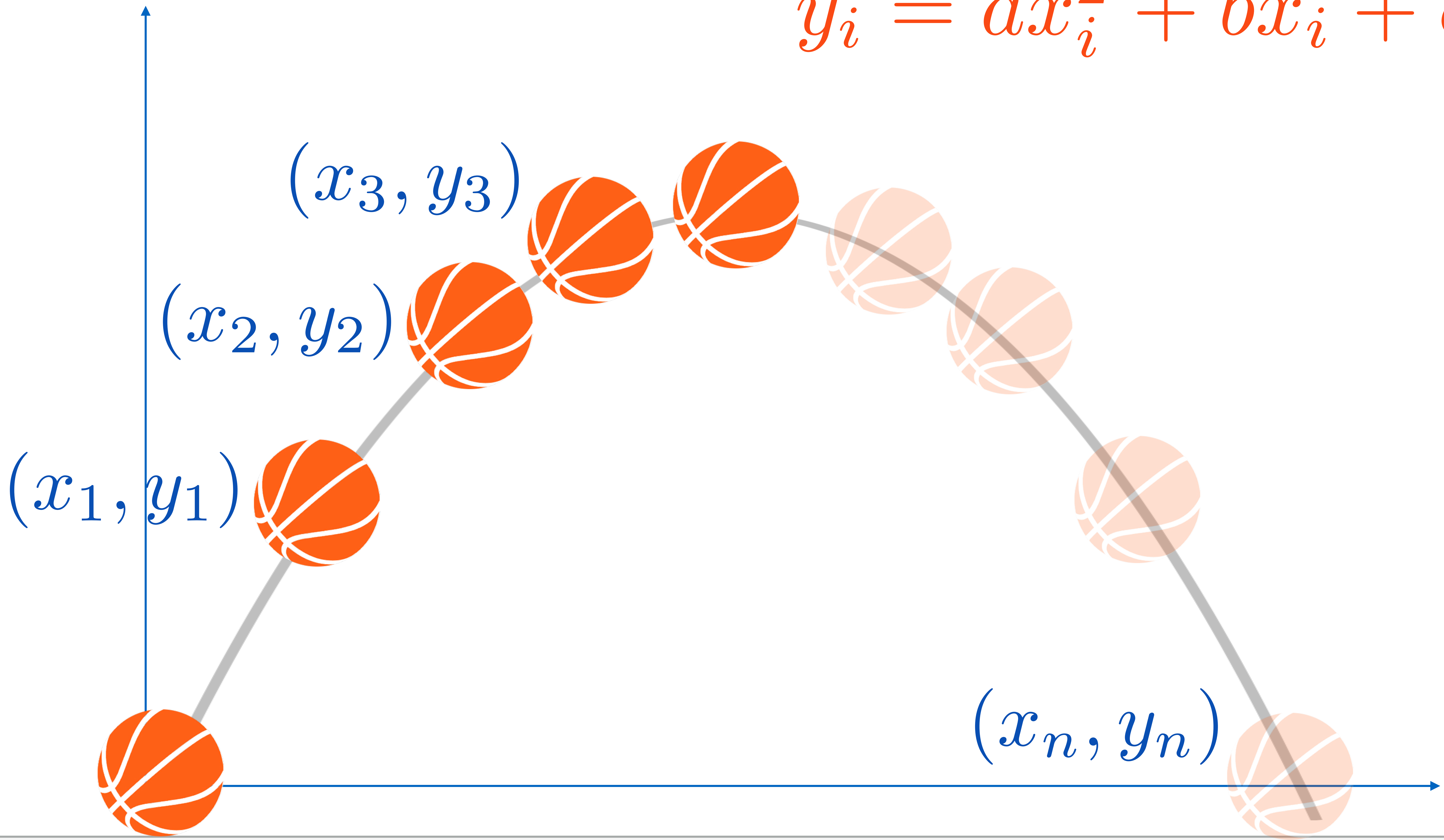
Recall – afterthoughts

What if we have more than three data points?

- which ones should we use?
  - only 3?
  - all of them?
  - some (which?) subset?

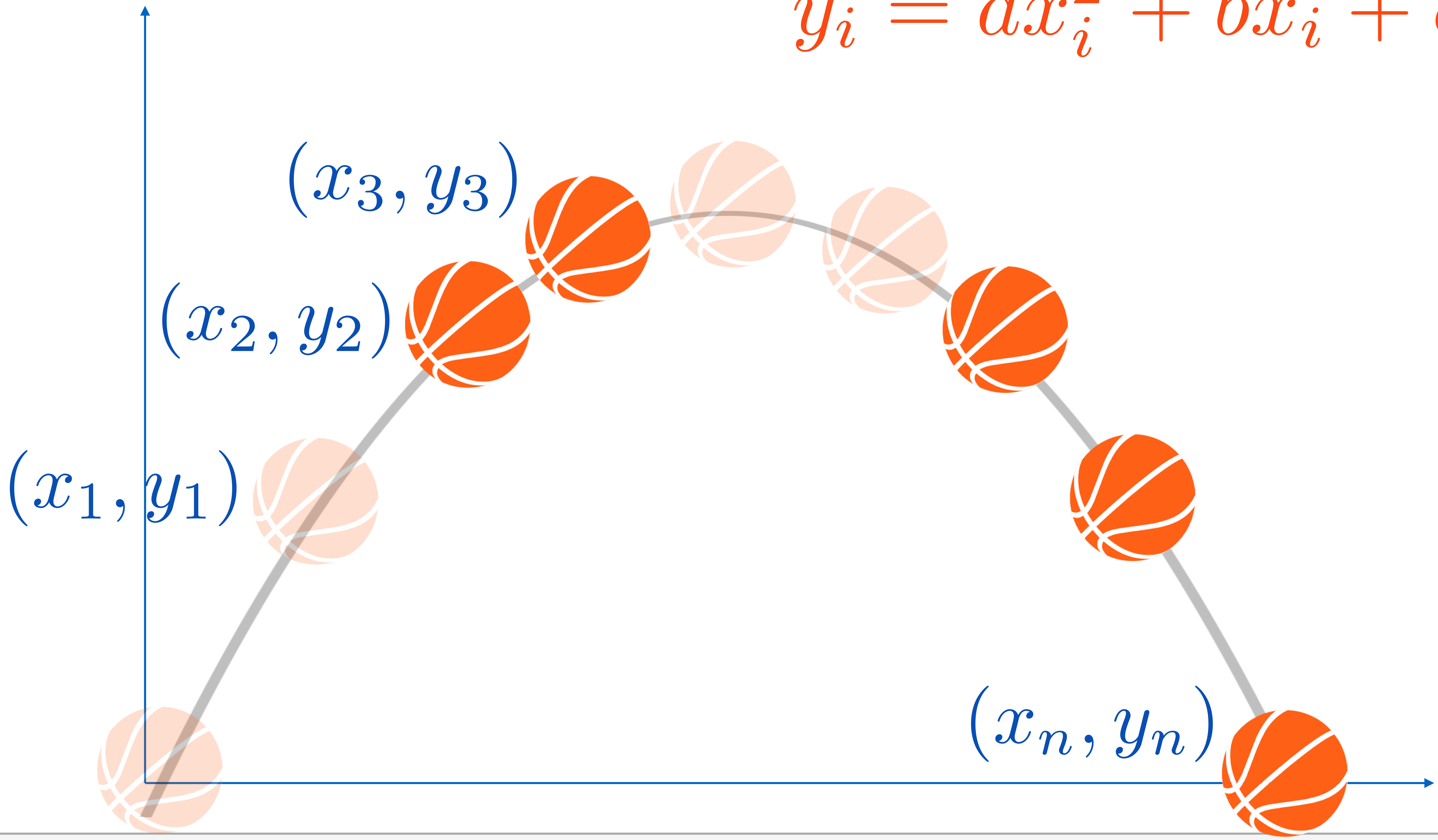
Recall – afterthoughts

$$y_i = ax_i^2 + bx_i + c$$



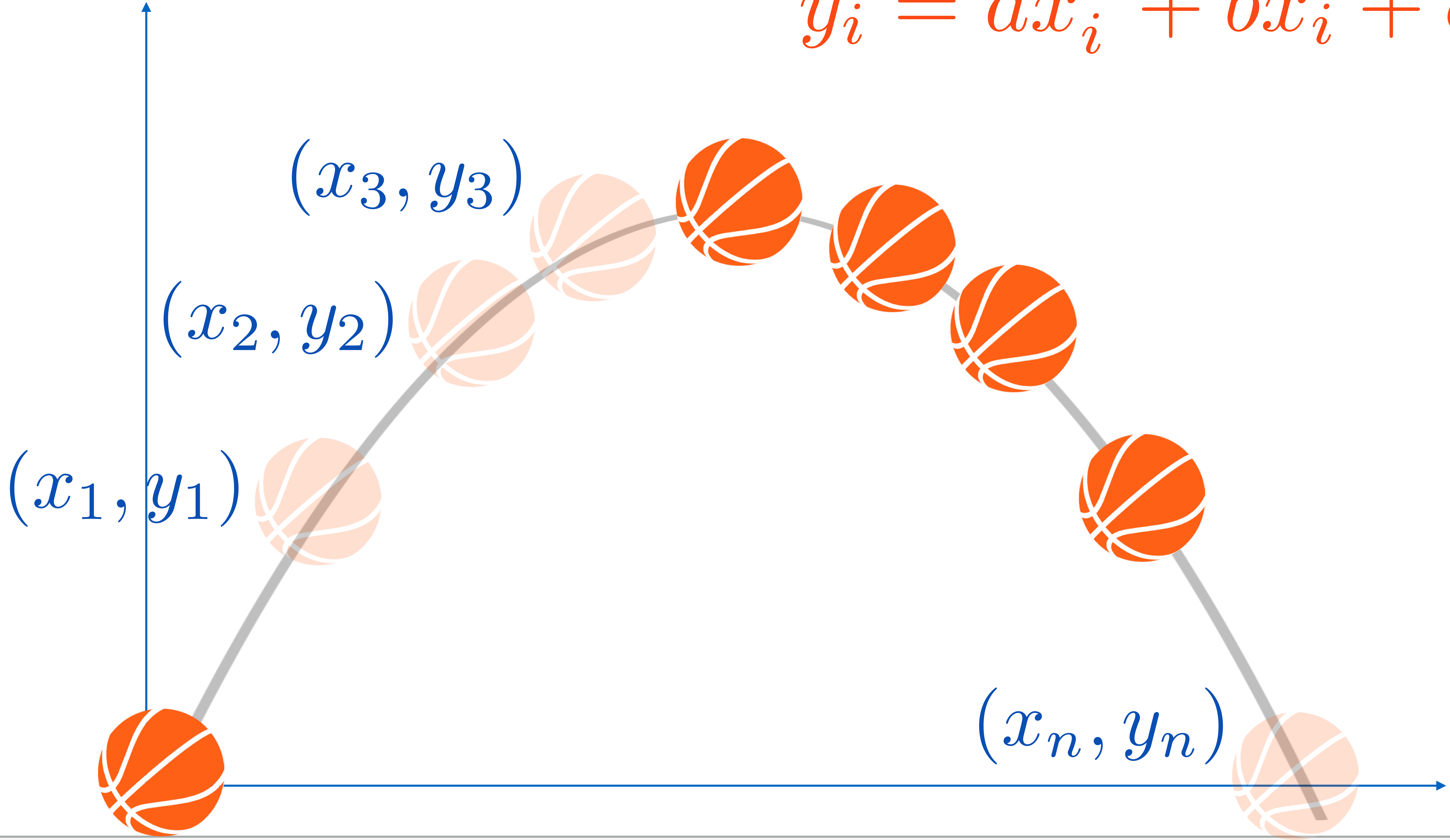
Recall – afterthoughts

$$y_i = ax_i^2 + bx_i + c$$



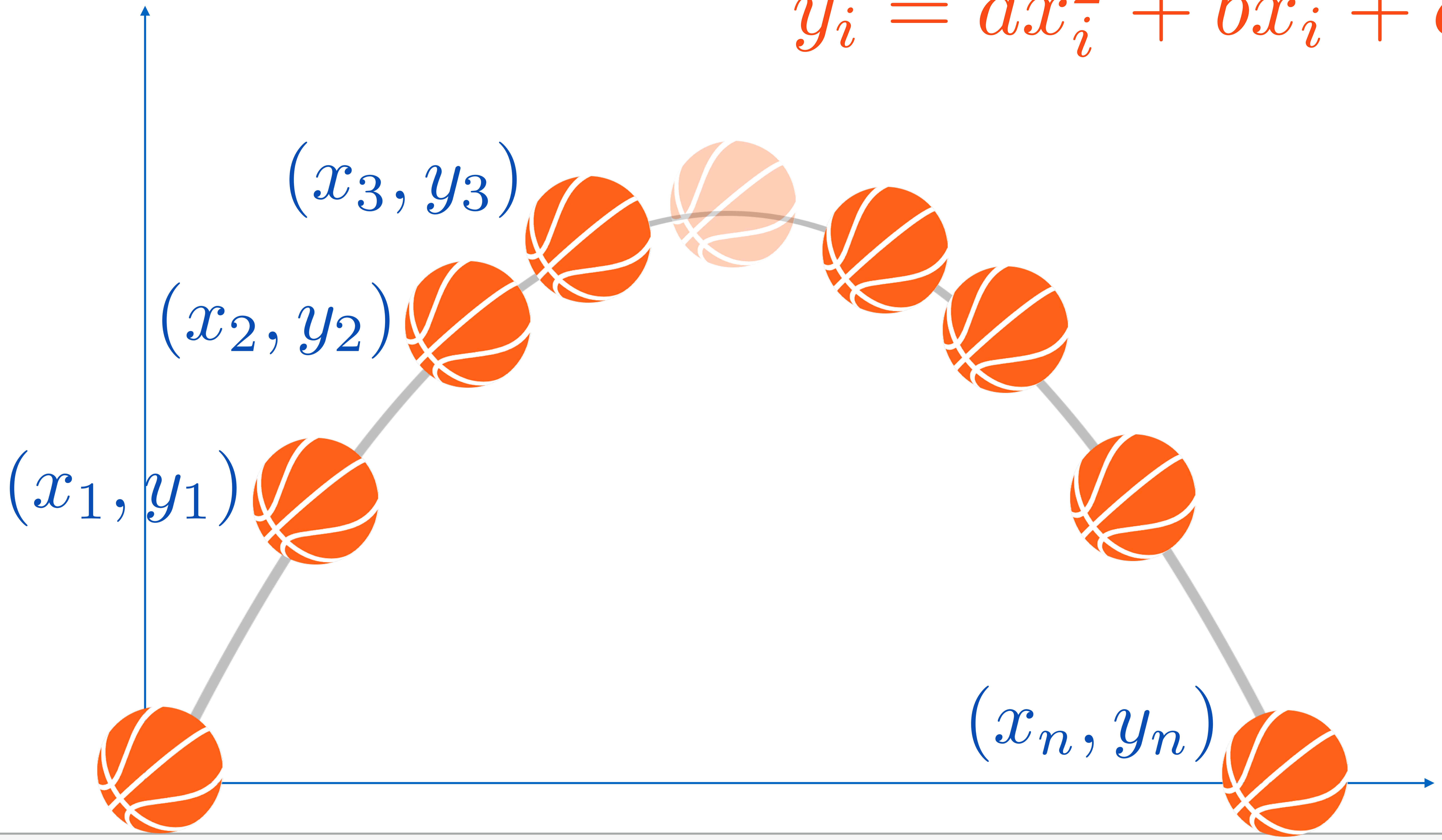
Recall – afterthoughts

$$y_i = ax_i^2 + bx_i + c$$

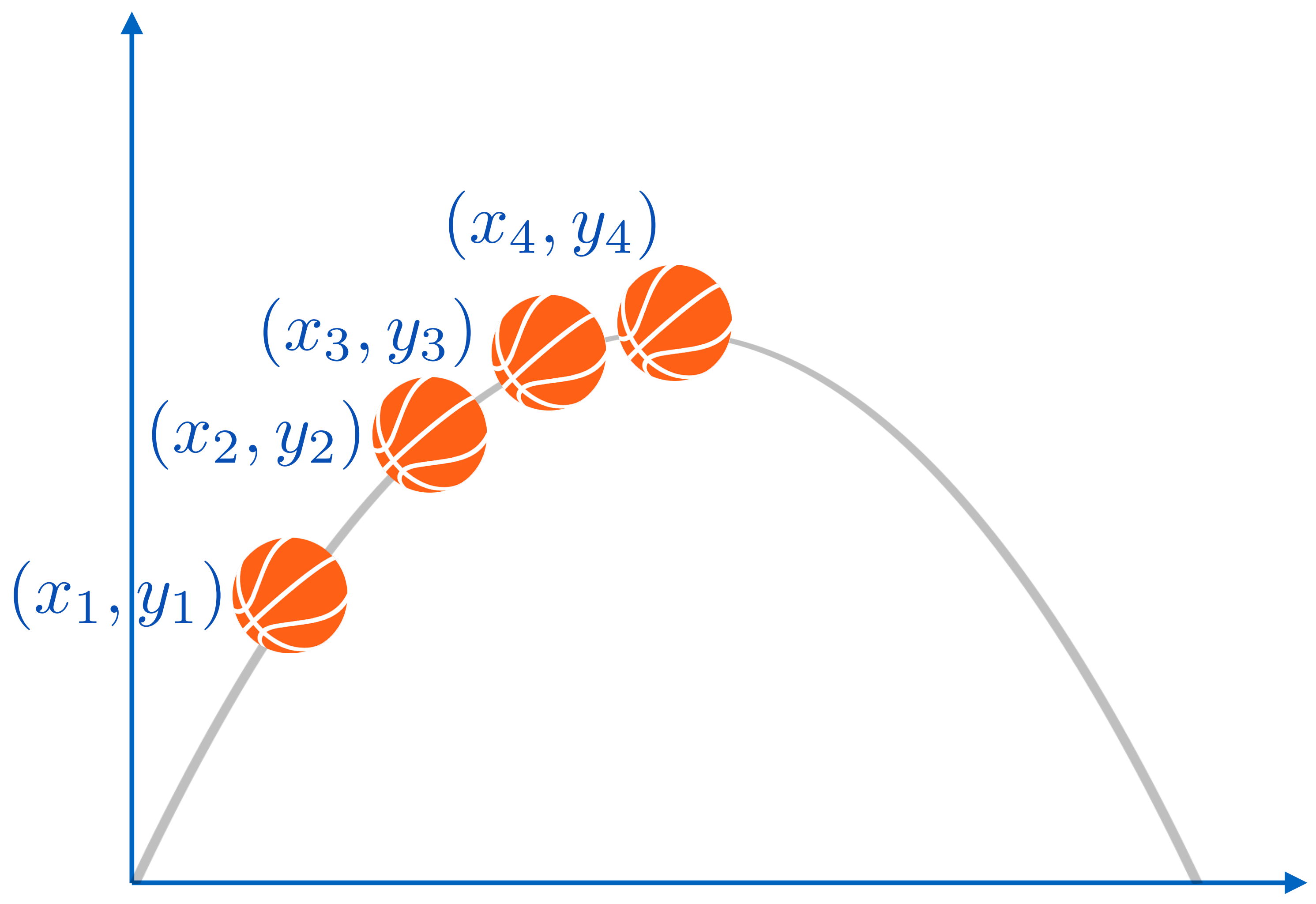


Recall – afterthoughts

$$y_i = ax_i^2 + bx_i + c$$



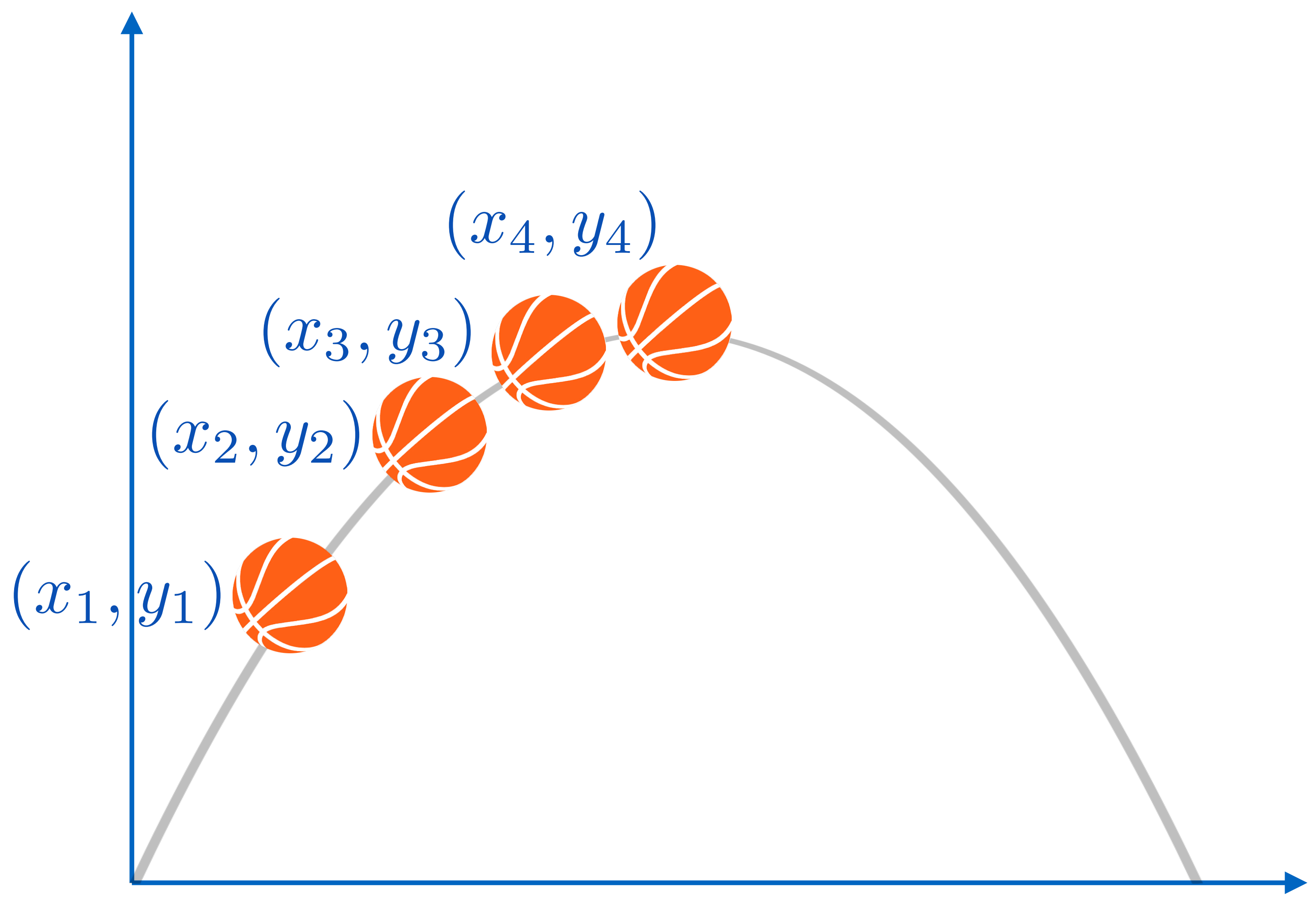
Recall – afterthoughts



$$y_1 = ax_1^2 + bx_1 + c$$
$$y_2 = ax_2^2 + bx_2 + c$$
$$y_3 = ax_3^2 + bx_3 + c$$
$$y_4 = ax_4^2 + bx_4 + c$$

over-constrained

Recall – afterthoughts



$$
\begin{aligned}
y_1 &= ax_1^2 + bx_1 + c \\
y_2 &= ax_2^2 + bx_2 + c \\
y_3 &= ax_3^2 + bx_3 + c \\
y_4 &= ax_4^2 + bx_4 + c
\end{aligned}
$$

over-constrained:
three vectors in 4-D space
but need four vectors to form
a complete basis

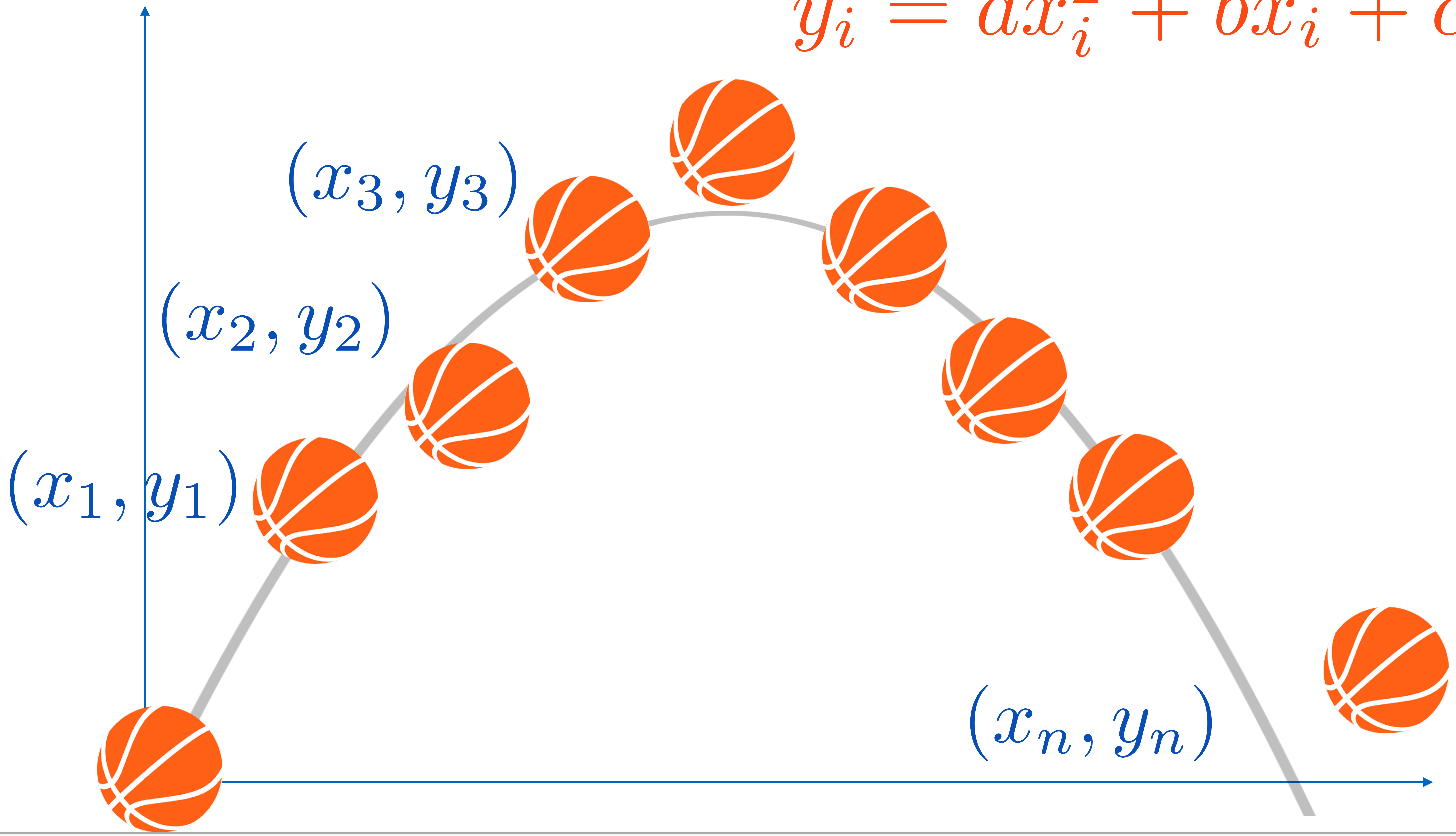
Recall – afterthoughts

What if we have more than three data points?

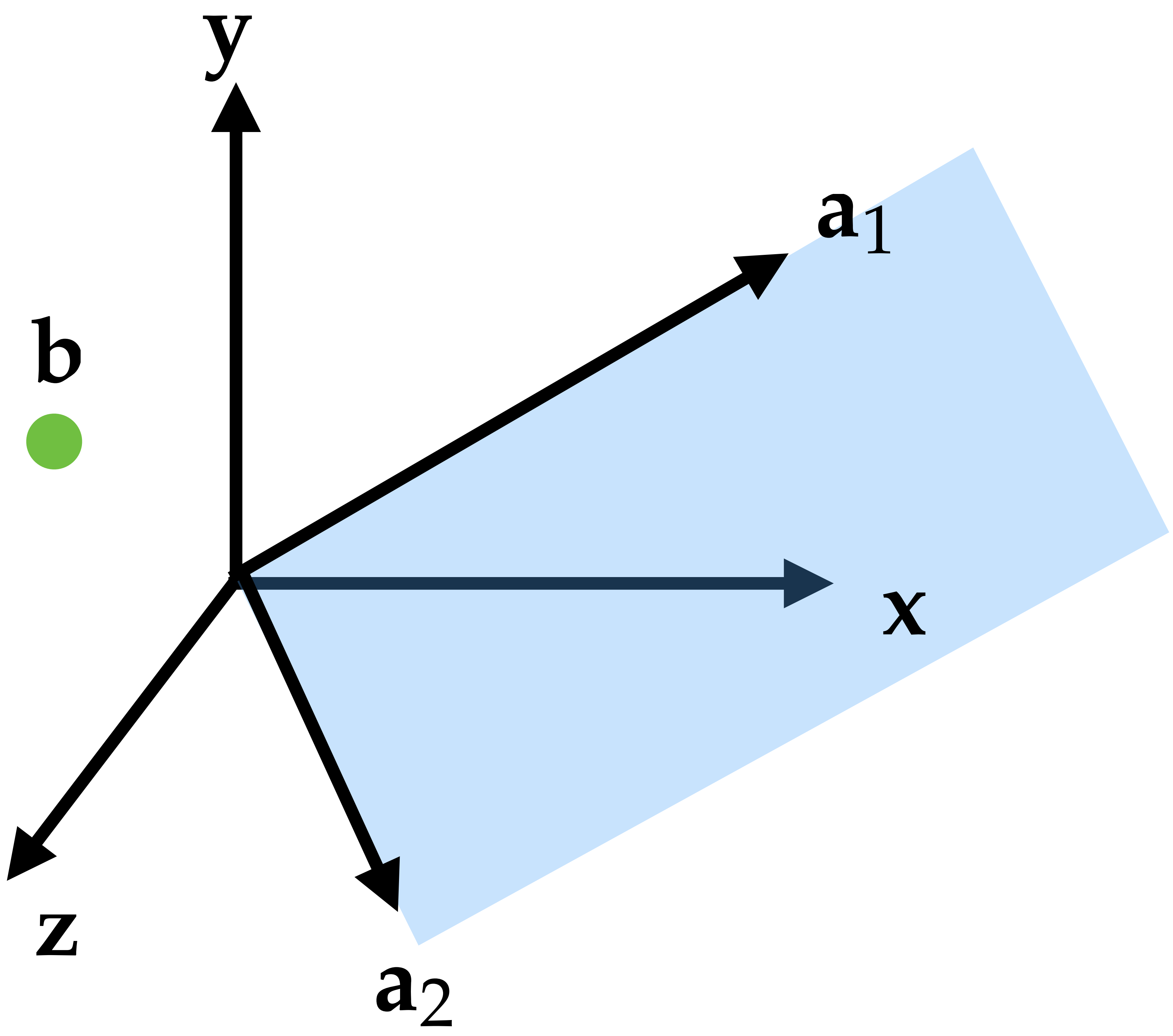
- which ones should we use?
- what if they don't "match" (i.e., respect the same model instance)?

Recall – afterthoughts

$$y_i = ax_i^2 + bx_i + c$$



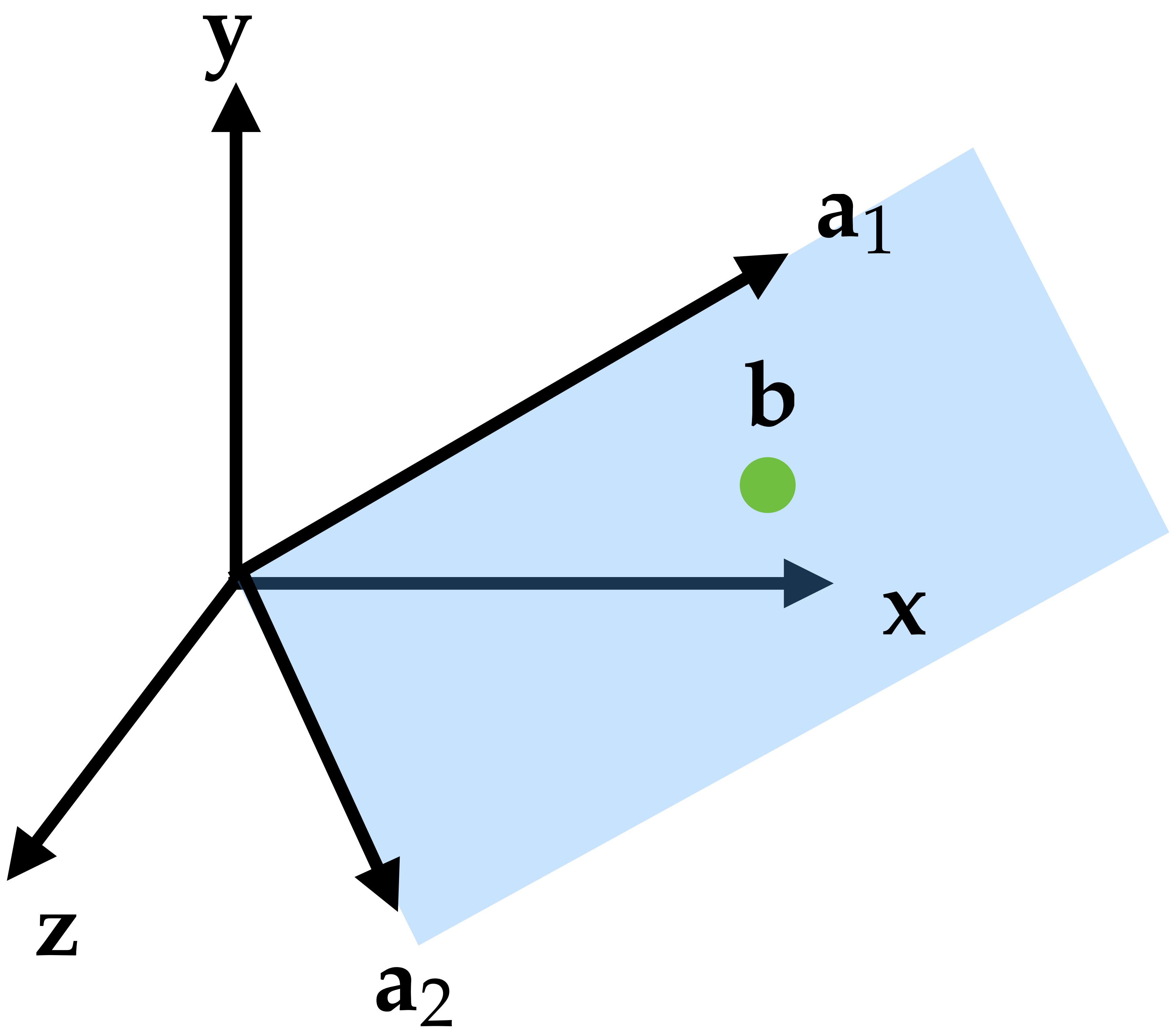
"Tall" / overdetermined systems ($m > n$)



$$\mathbf{A} = \begin{pmatrix} | & | \\ \mathbf{a}_1 & \mathbf{a}_2 \\ | & | \end{pmatrix} \Bigg\} \text{ 3 rows}$$

For almost* all values of $\mathbf{b}$, $\mathbf{Ax} = \mathbf{b}$ *is **not** solvable*

"Tall" / overdetermined systems ($m > n$)

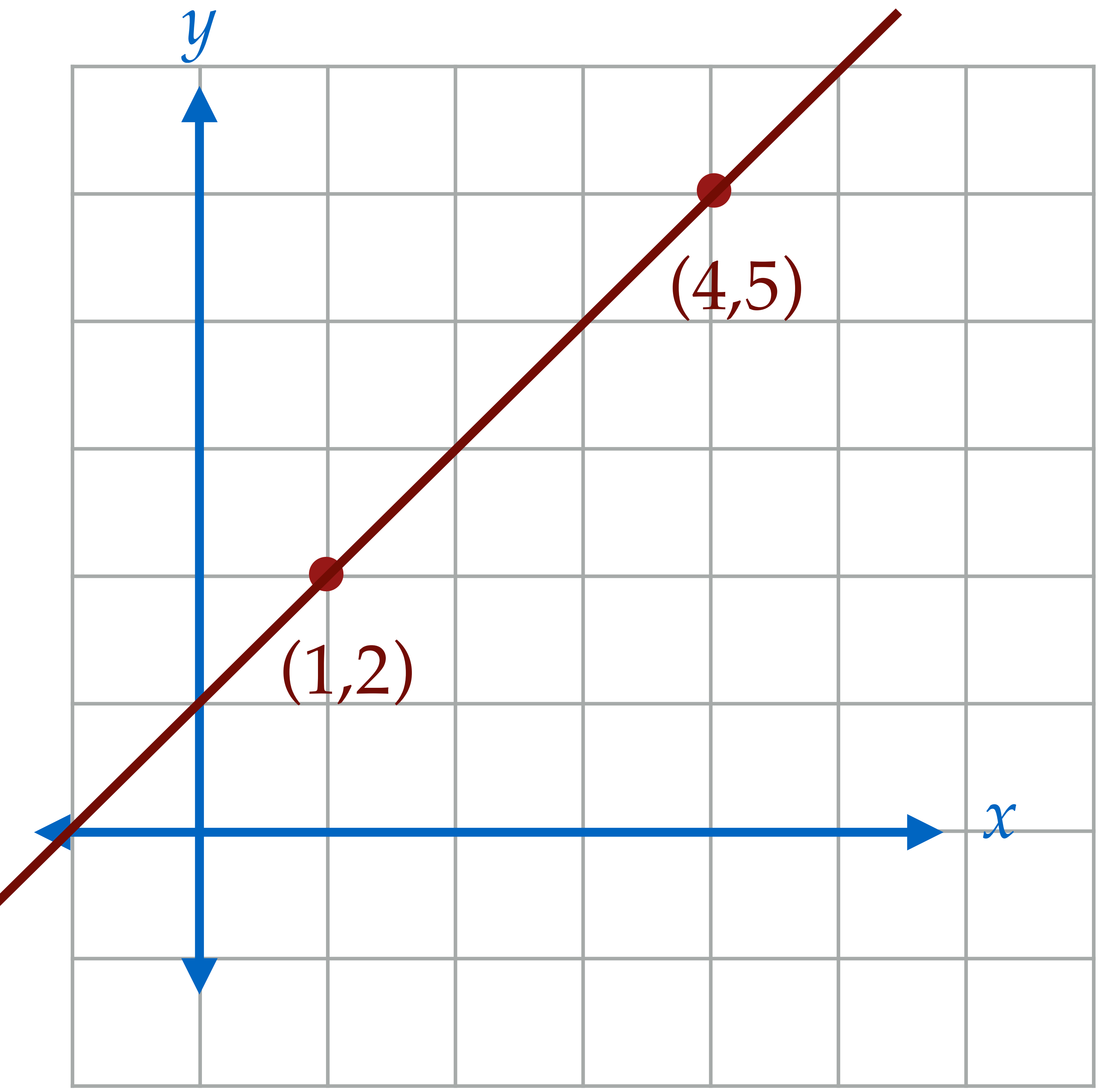


$$\mathbf{A} = \left.\begin{pmatrix} | & | \\ \mathbf{a}_1 & \mathbf{a}_2 \\ | & | \end{pmatrix}\right\} \text{3 rows}$$

When we have more constraints/equations than unknowns:

- "tall", overdetermined, over-constrained

More Points Than we Need… so what?



Geometric picture

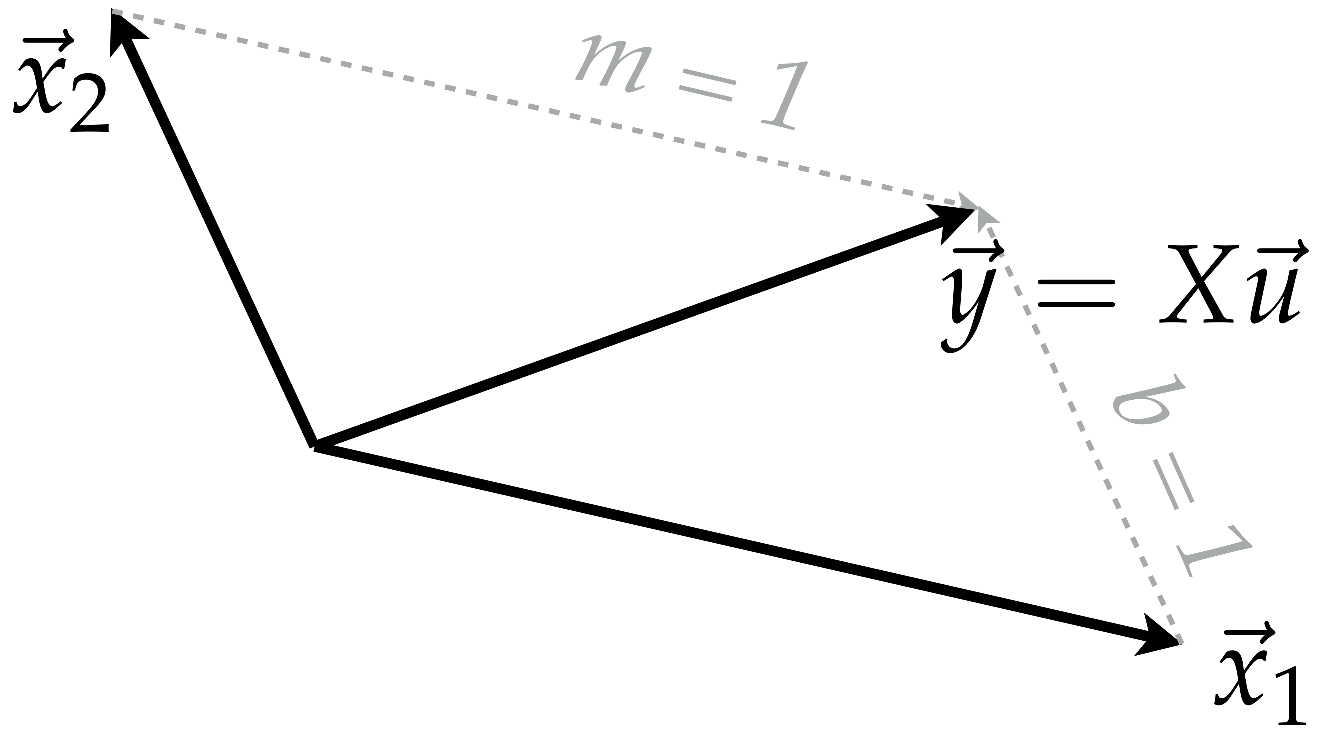
$$m = 1$$
$$b = 1$$

$$\begin{pmatrix} 1 \\ 4 \end{pmatrix} m + \begin{pmatrix} 1 \\ 1 \end{pmatrix} b = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

$$\vec{x}_1 m + \vec{x}_2 b = \vec{y}$$

$$\begin{pmatrix} 1 & 1 \\ 4 & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

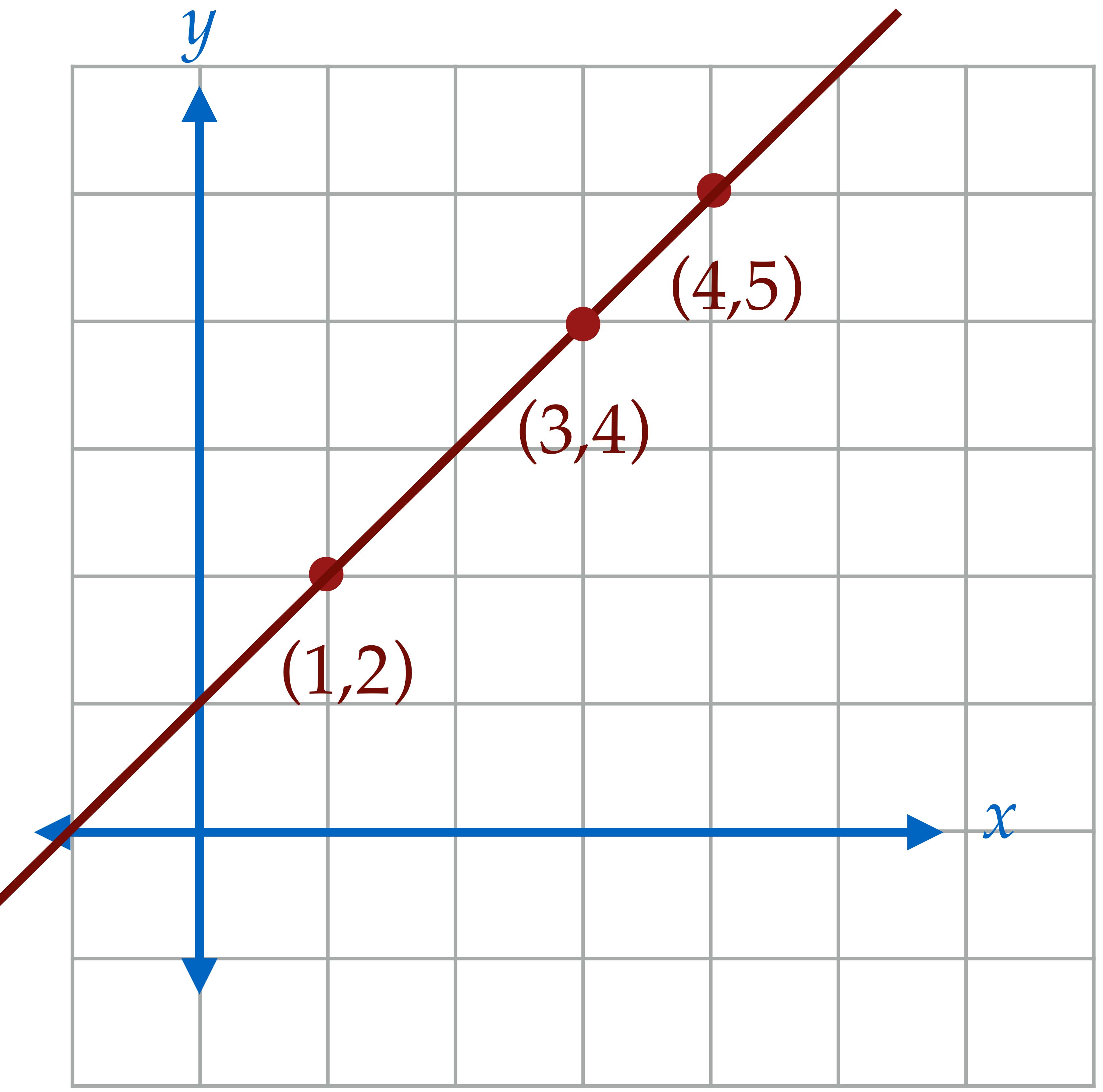
$$X\vec{u} = \vec{y}$$



(2D plane)

Vector picture

Over-constrained Fit – very lucky



Geometric picture

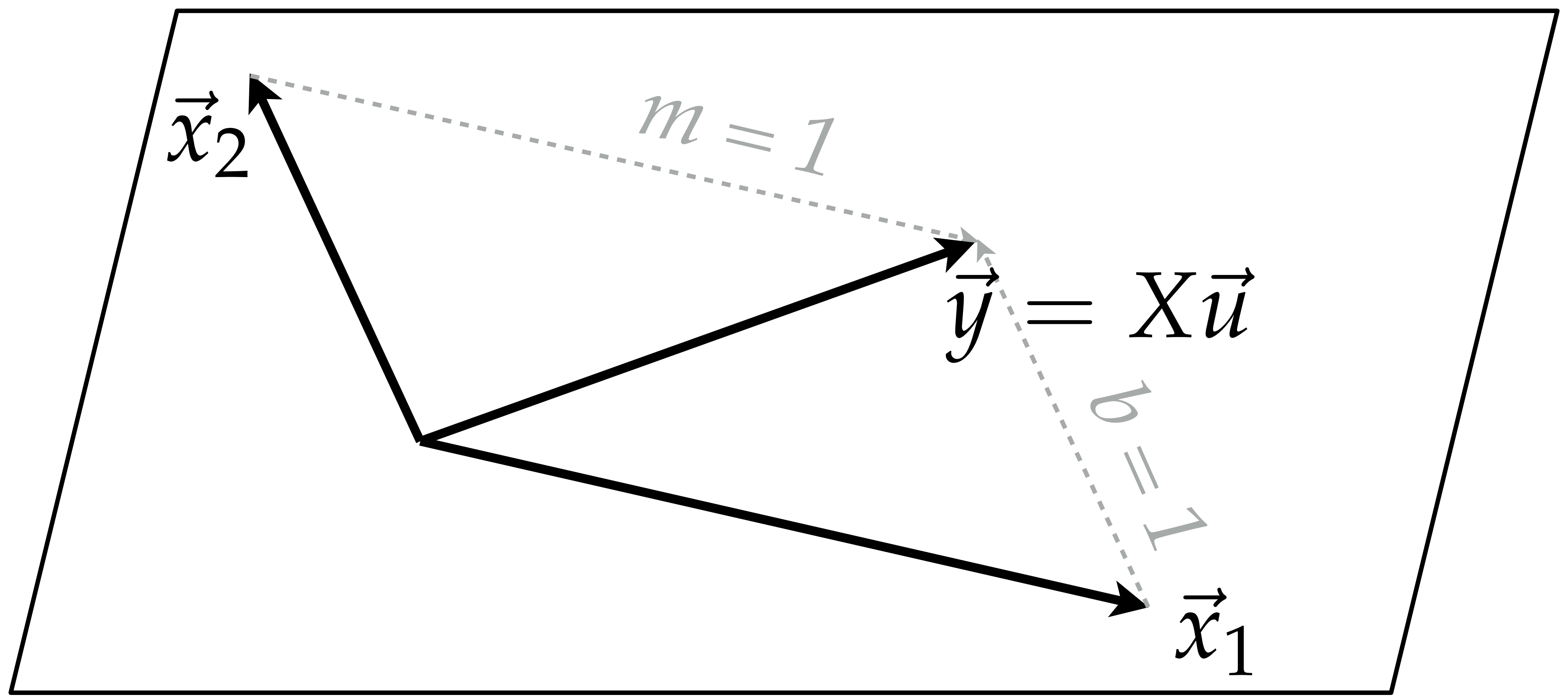
$$m = 1$$
$$b = 1$$

$$\begin{pmatrix} 1 \\ 4 \\ 3 \end{pmatrix} m + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} b = \begin{pmatrix} 2 \\ 5 \\ 4 \end{pmatrix}$$

$$\vec{x}_1 m + \vec{x}_2 b = \vec{y}$$

$$\begin{pmatrix} 1 & 1 \\ 4 & 1 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 4 \end{pmatrix}$$

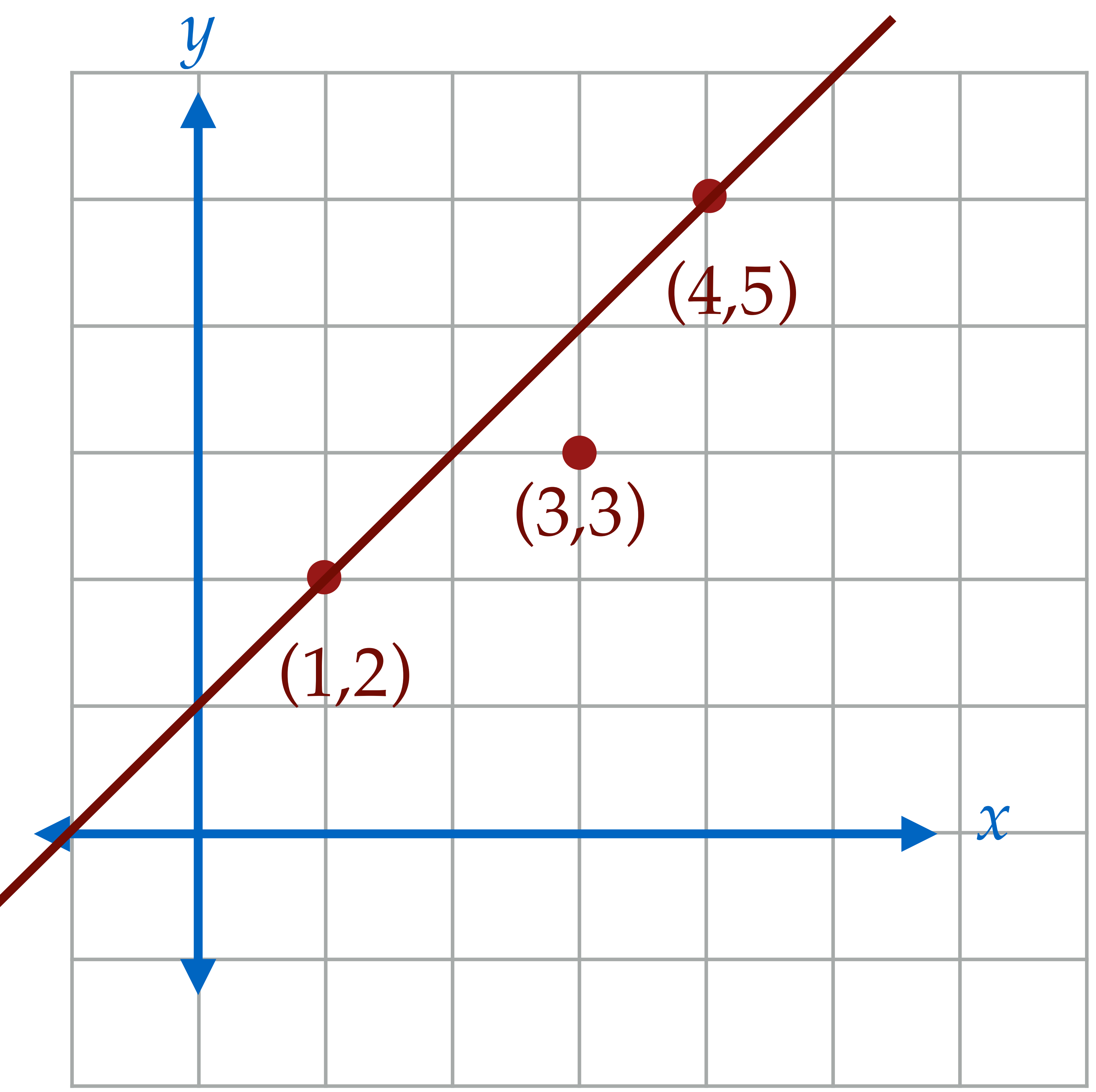
$$X\vec{u} = \vec{y}$$



(2D plane in 3D)

Vector picture

Over-constrained Fit – general case



Geometric picture

???

$$m = ?$$

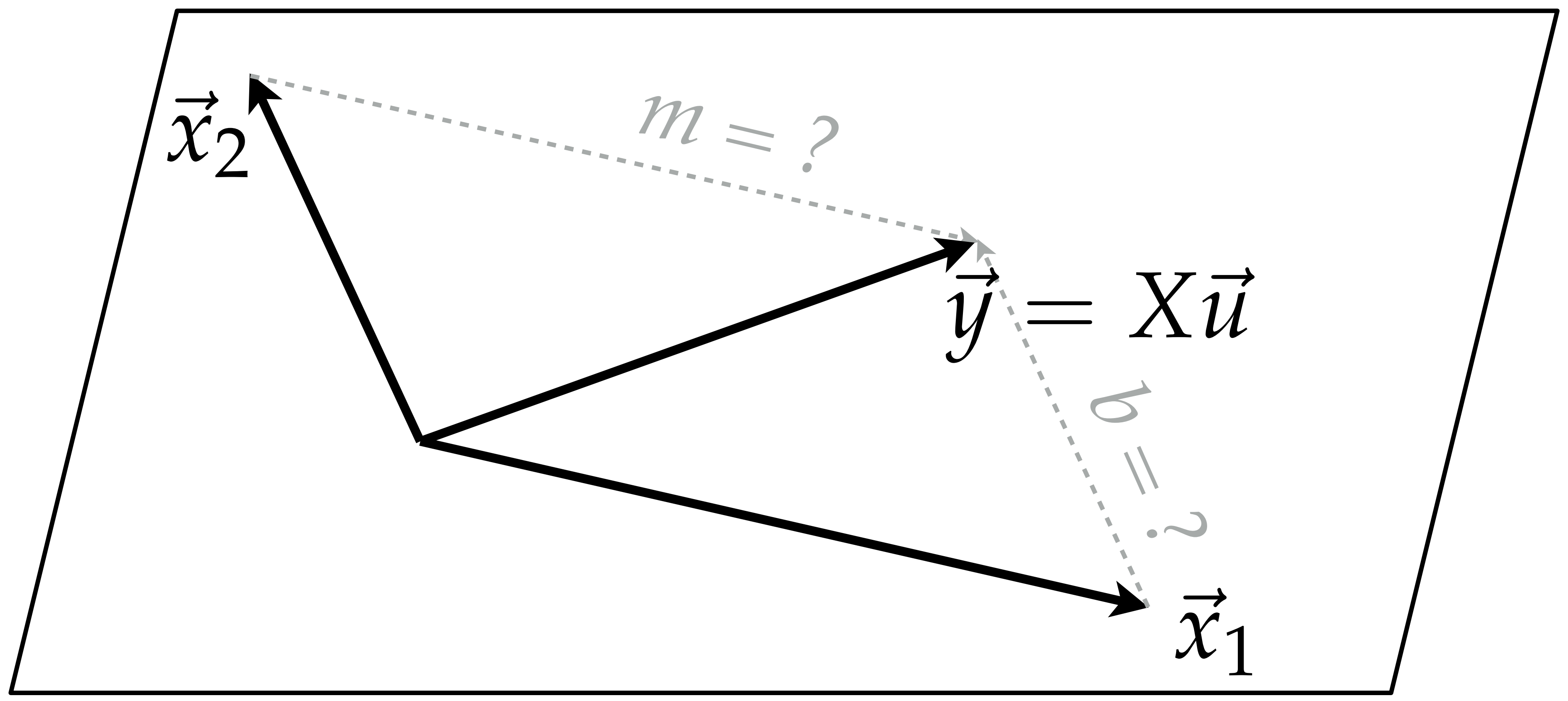
$$b = ?$$

$$\begin{pmatrix}1\\4\\3\end{pmatrix} m + \begin{pmatrix}1\\1\\1\end{pmatrix} b = \begin{pmatrix}2\\5\\3\end{pmatrix}$$

$$\vec{x}_1 m + \vec{x}_2 b = \vec{y}$$

$$\begin{pmatrix}1 & 1\\4 & 1\\3 & 1\end{pmatrix} \begin{pmatrix}m\\b\end{pmatrix} = \begin{pmatrix}2\\5\\3\end{pmatrix}$$

$$X\vec{u} = \vec{y}$$

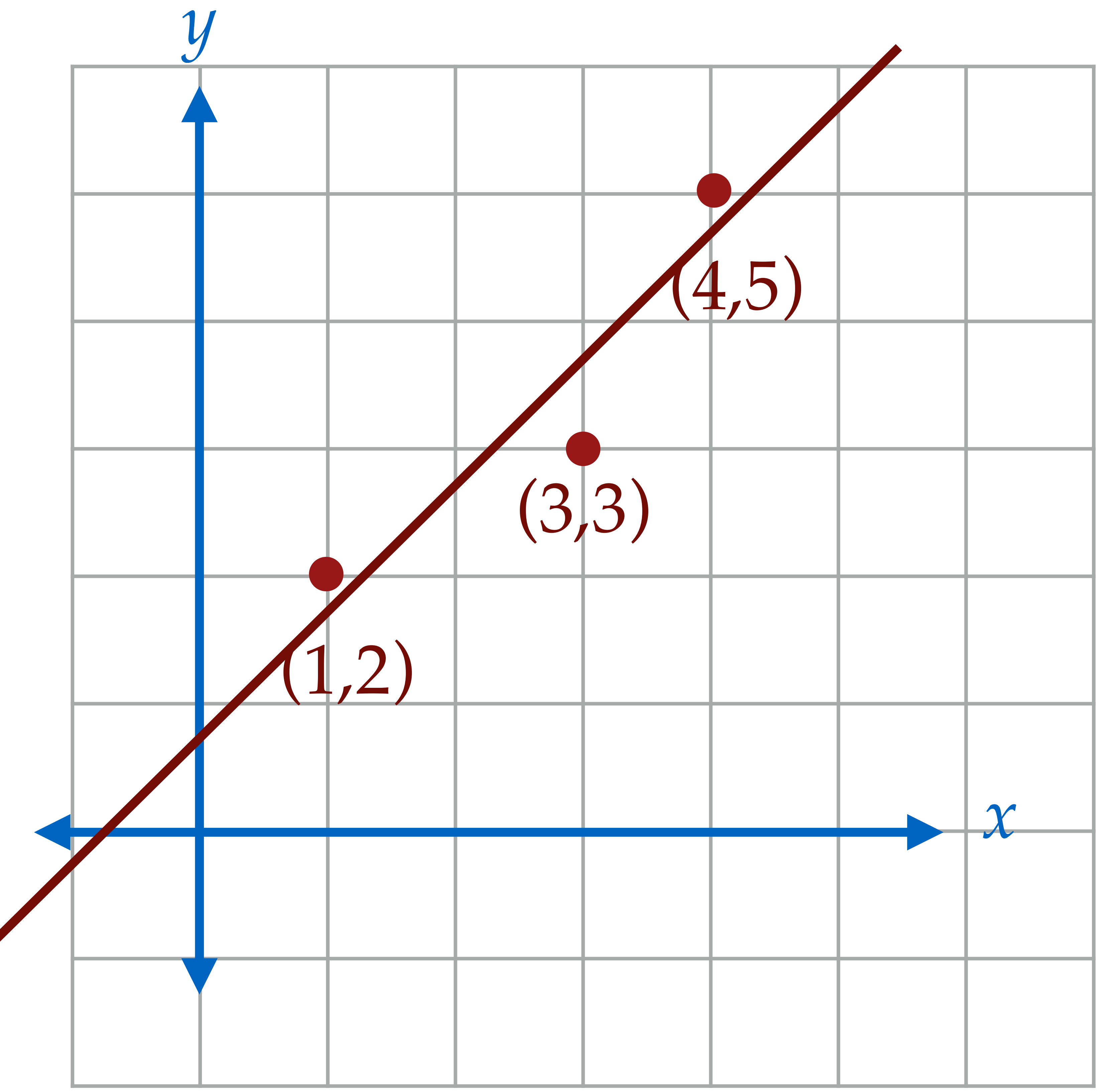


(2D plane in 3D)

Vector picture

???

Over-constrained Fit – general case



Geometric picture

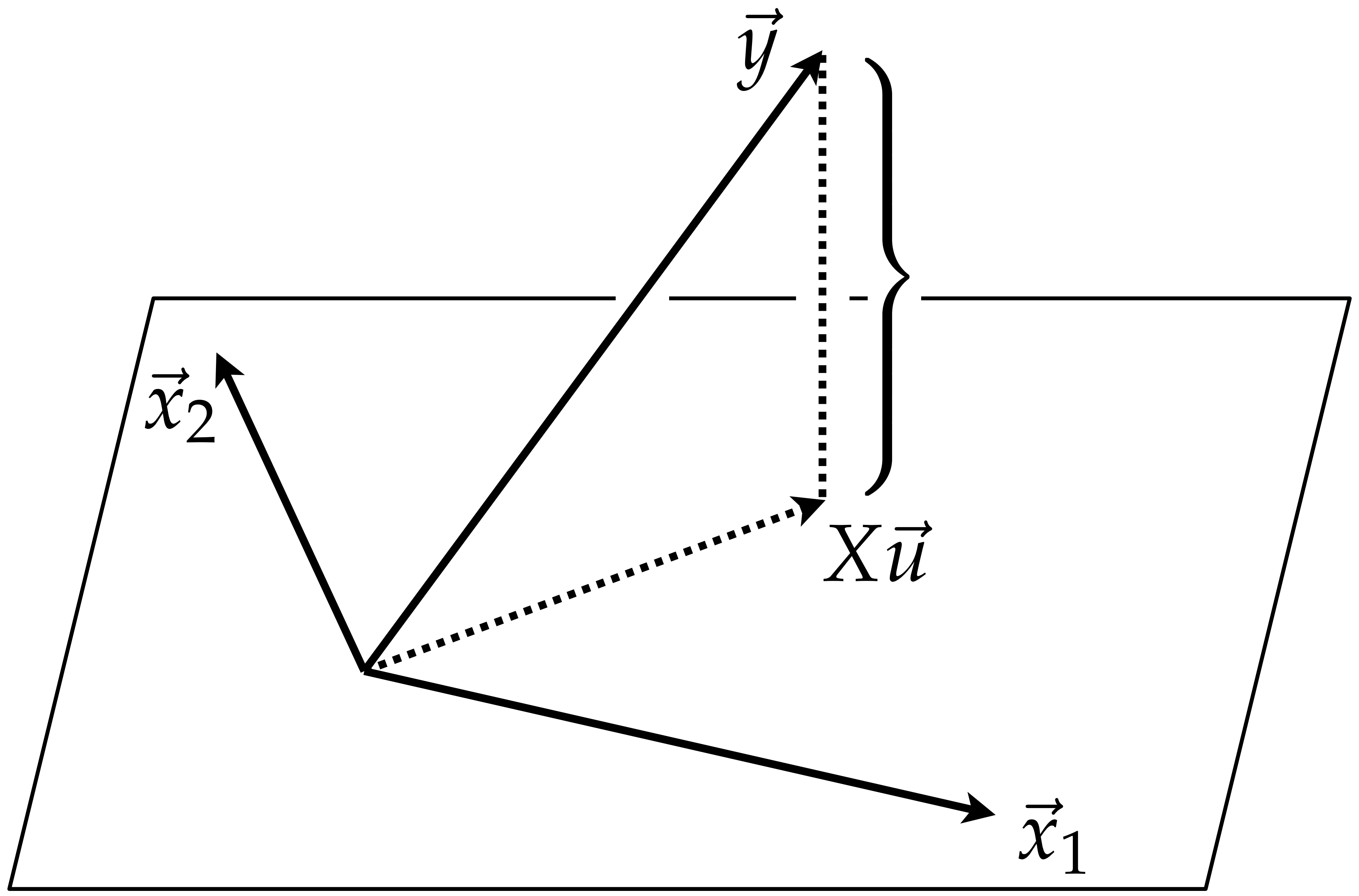
$$m = ?$$
$$b = ?$$

$$\begin{pmatrix}1\\4\\3\end{pmatrix} m + \begin{pmatrix}1\\1\\1\end{pmatrix} b \neq \begin{pmatrix}2\\5\\3\end{pmatrix}$$

$$\vec{x}_1 m + \vec{x}_2 b \neq \vec{y}$$

$$\begin{pmatrix}1 & 1\\4 & 1\\3 & 1\end{pmatrix} \begin{pmatrix}m\\b\end{pmatrix} \neq \begin{pmatrix}2\\5\\3\end{pmatrix}$$

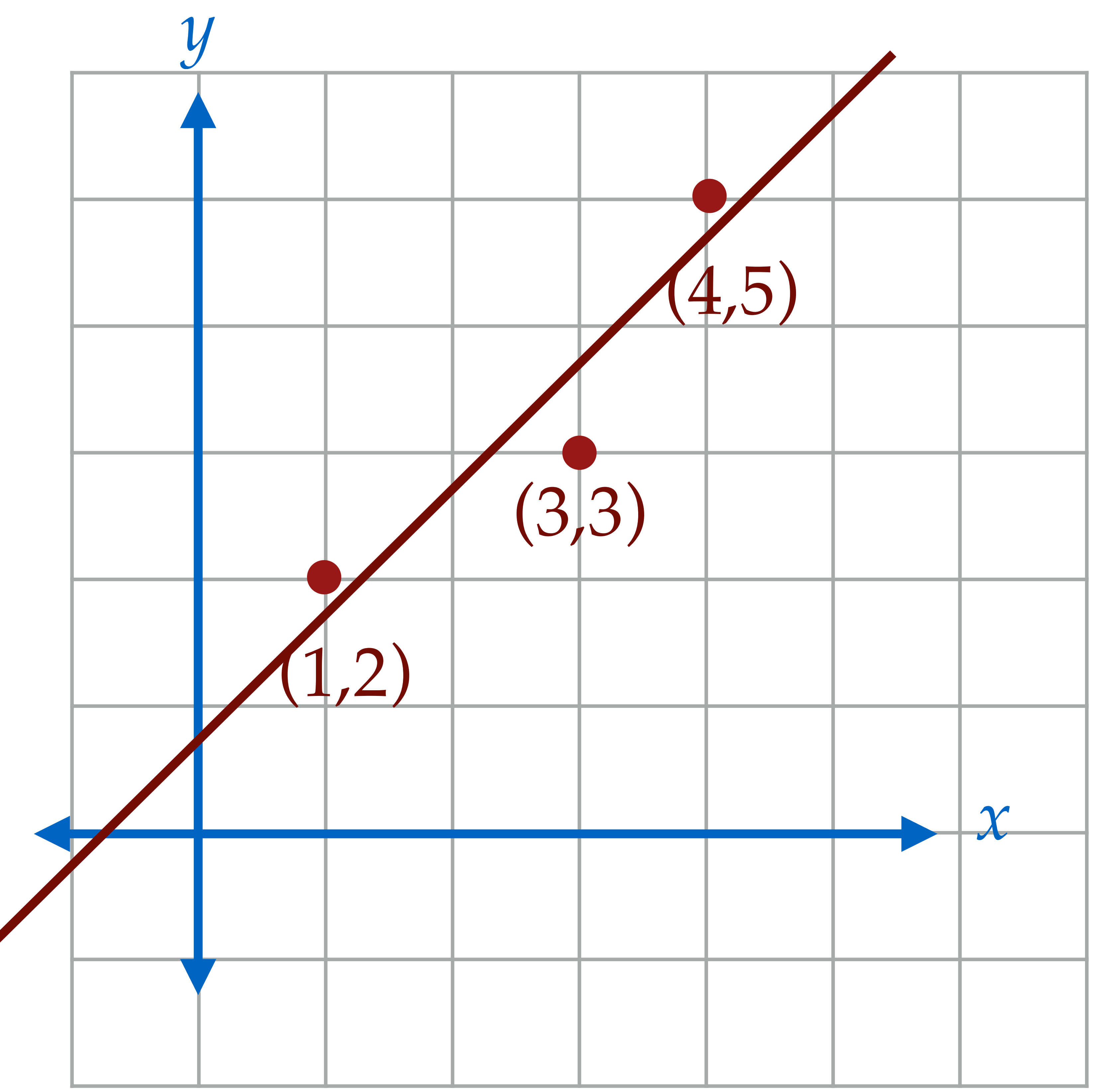
$$X\vec{u} \neq \vec{y}$$



(2D plane in 3D)

Vector picture

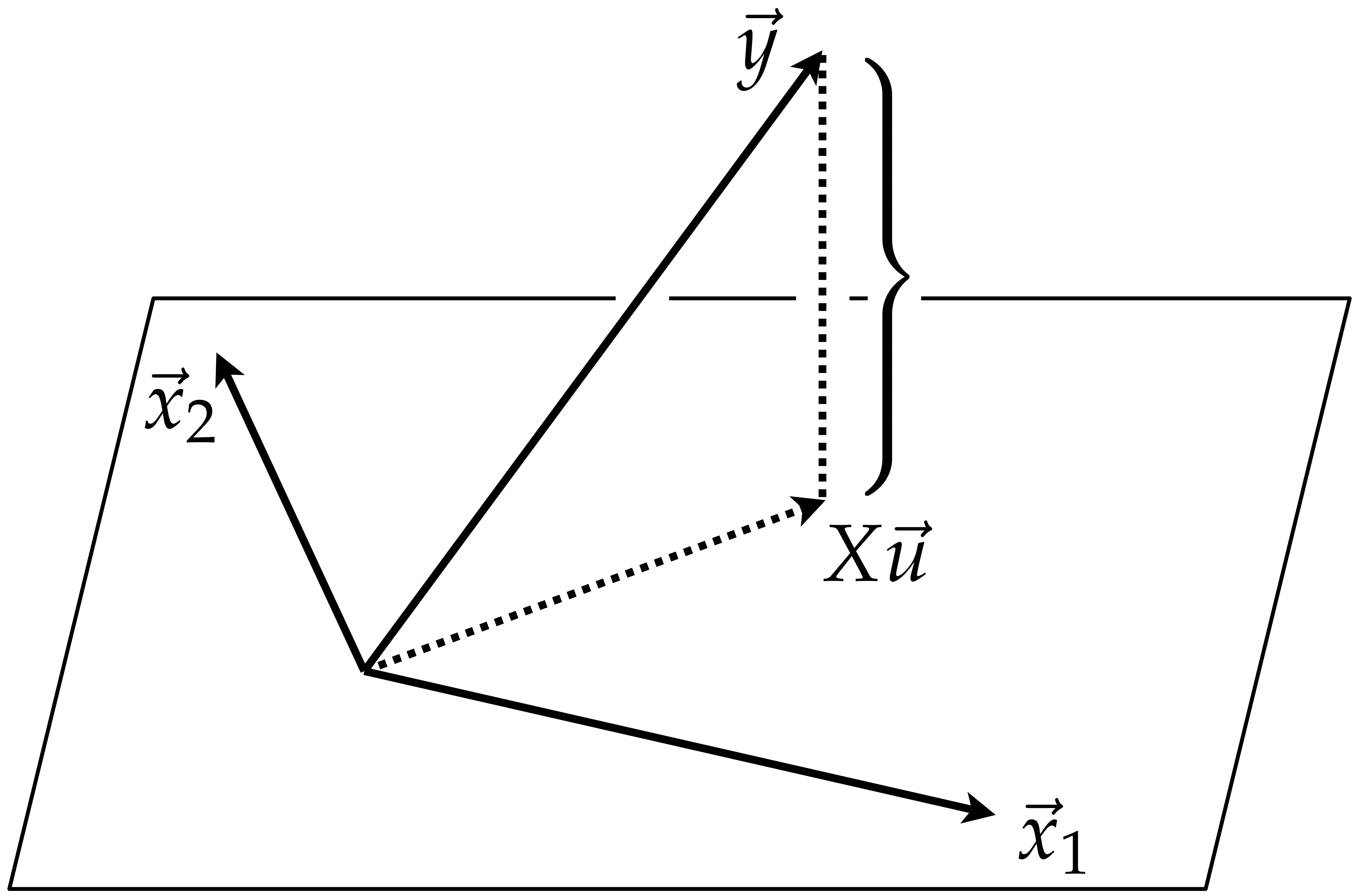
Over-constrained Fit – general case



Geometric picture

$$X\vec{u} \neq \vec{y}$$

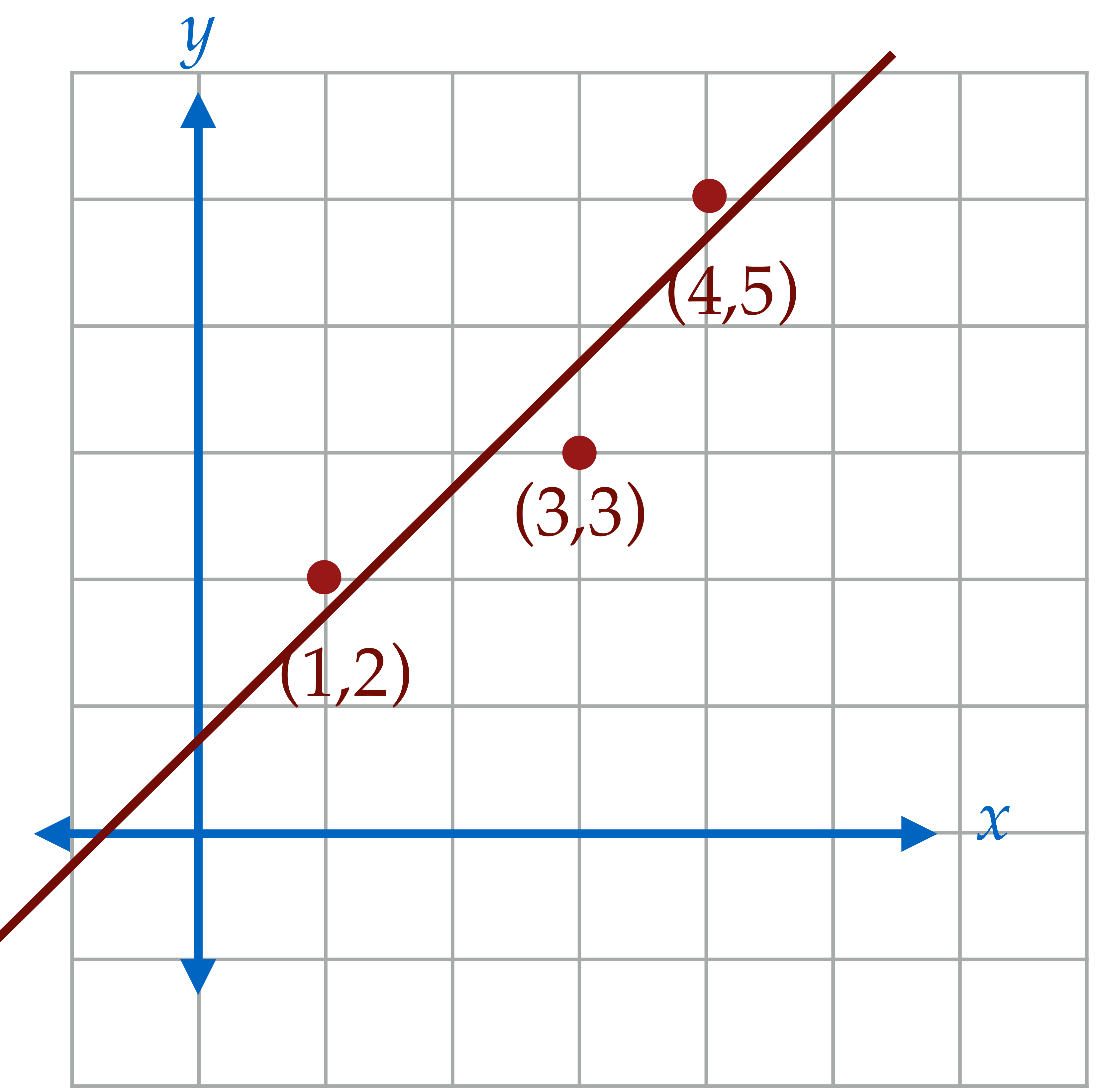
$$X\vec{u} - \vec{y} = \vec{\Delta}$$



(2D plane in 3D)

Vector picture

Over-constrained Fit – general case

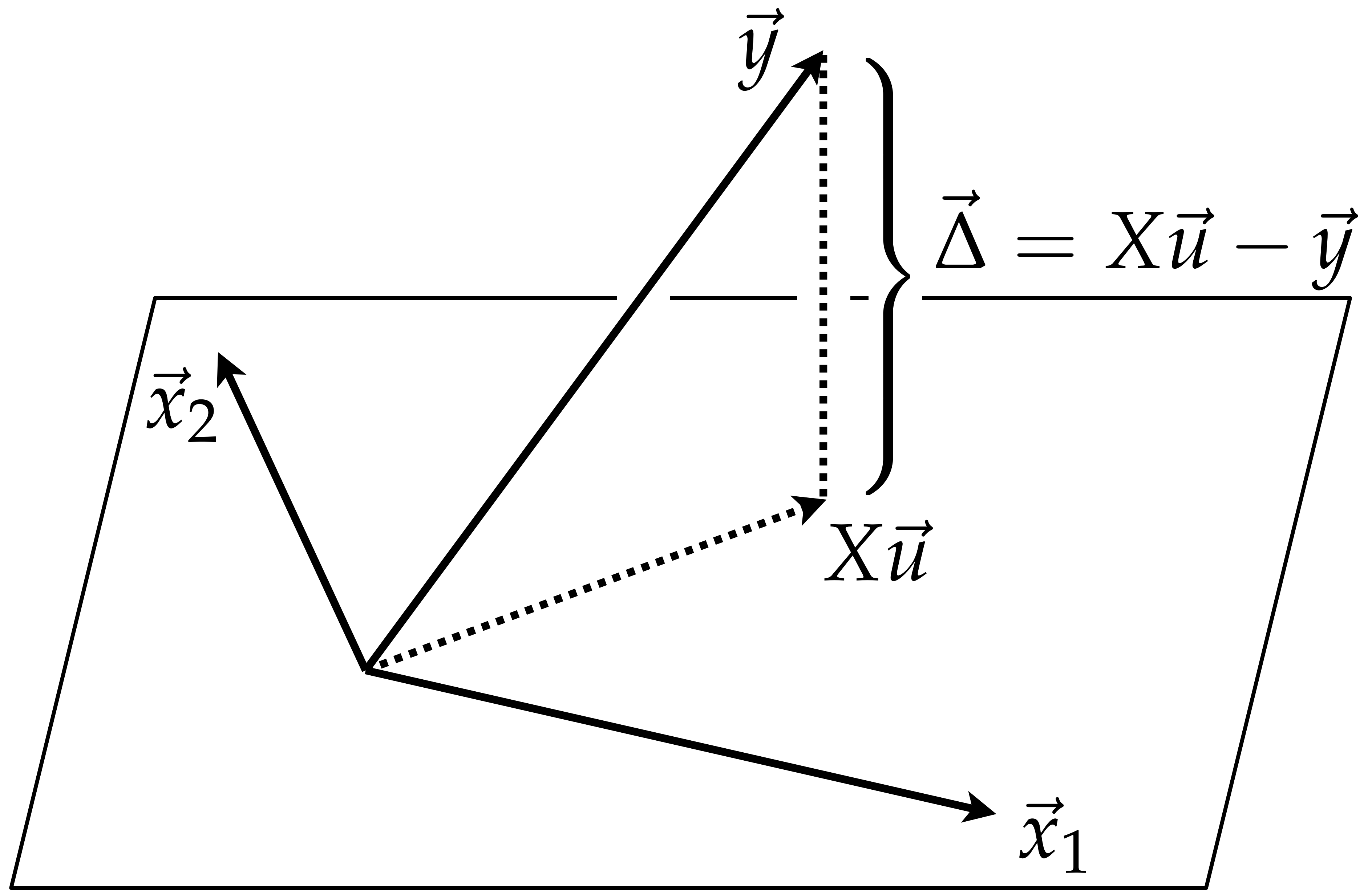


Geometric picture

$$X\vec{u} \neq \vec{y}$$

$$X\vec{u} - \vec{y} = \vec{\Delta}$$

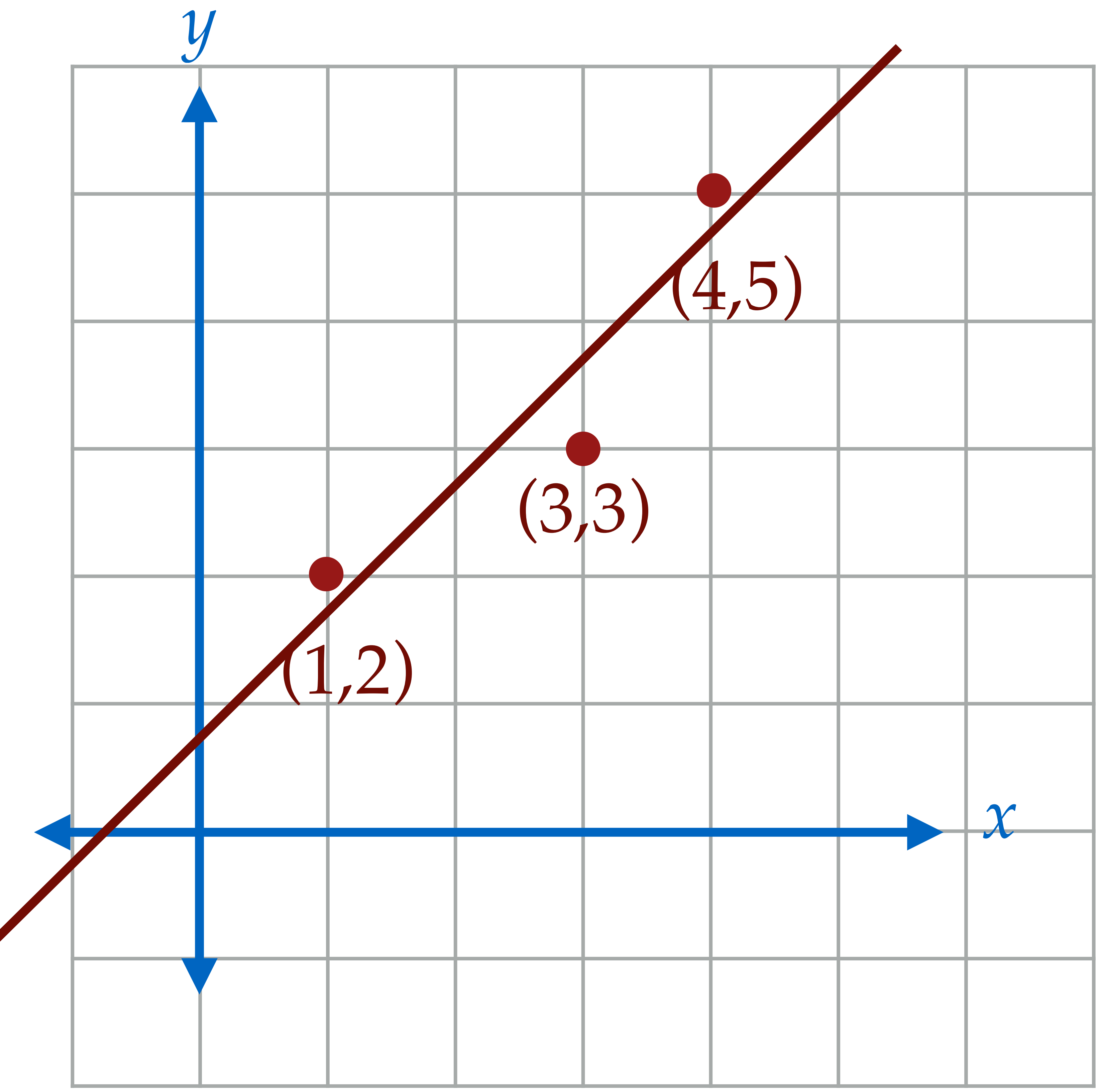
$$\min_{\vec{u}} \|X\vec{u} - \vec{y}\|^2$$



(2D plane in 3D)

Vector picture

Over-constrained Fit – general case

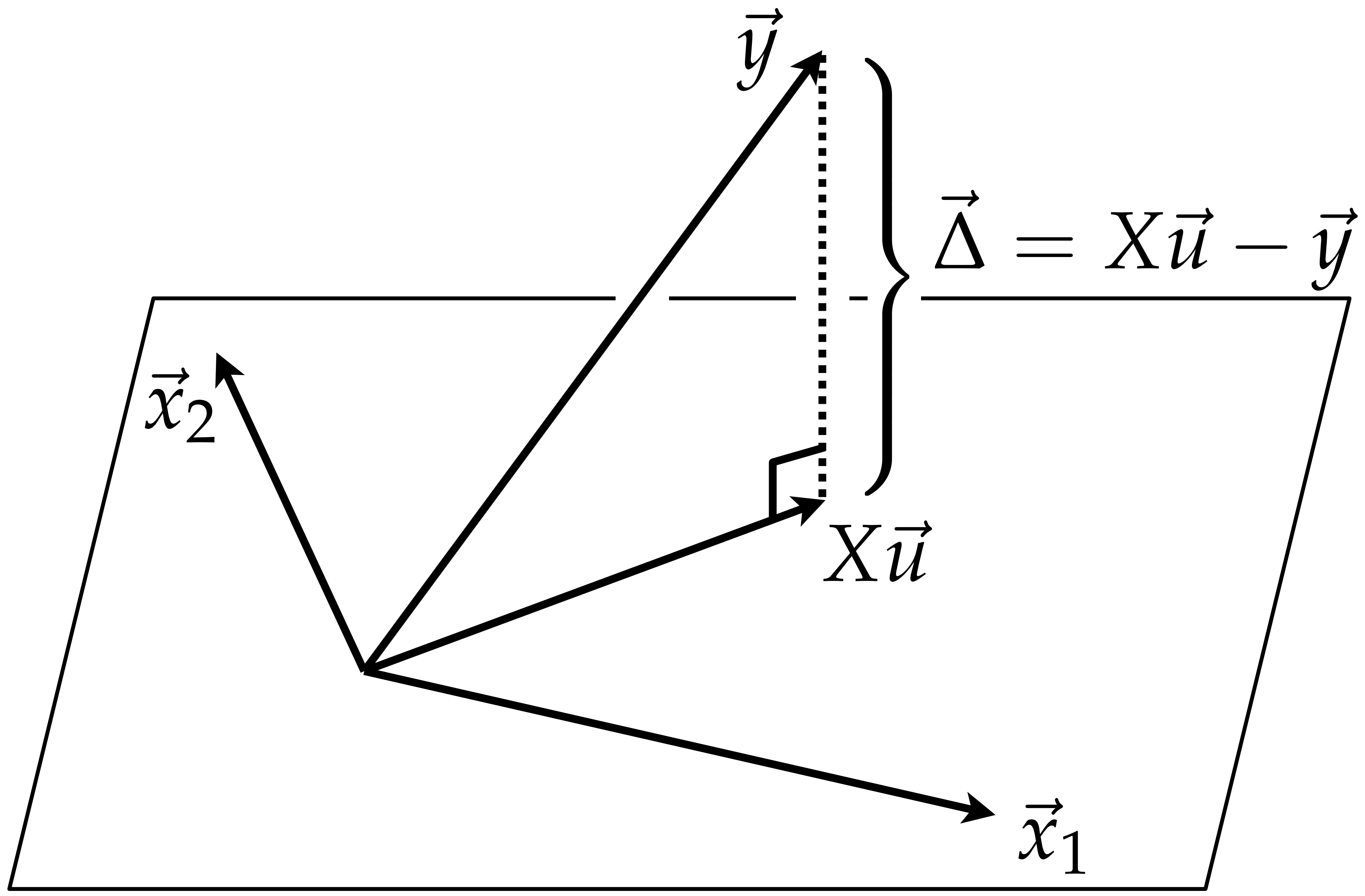


Geometric picture

$$X\vec{u} \neq \vec{y}$$

$$X\vec{u} - \vec{y} = \vec{\Delta}$$

$$\min_{\vec{u}} \|X\vec{u} - \vec{y}\|^2$$



(2D plane in 3D)

Vector picture

Least-squares Estimation

The squared error is defined as

$$\vec{\Delta} \equiv r(\vec{u}) \equiv E(\vec{u}) = \|X\vec{u} - y\|^2$$

- also referred to as the *squared residual*, *squared residual energy*, *least squared error,* or *least squares energy*
- as we'll see shortly, squaring the residual will actually simplify our solution

Least-squares Estimation

known known

$$E(\vec{u}) = \|X\vec{u} - y\|^2$$

unknown

$$E(\vec{u}) = \left\| \begin{pmatrix} \text{---} \; X(1,:) \; \text{---} \\ \text{---} \; X(2,:) \; \text{---} \\ \vdots \\ \text{---} \; X(n,:) \; \text{---} \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_p \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \right\|^2$$

n constraints

p unknowns $n \geq p$

Linear Least Squares

Our goal is to minimize the residual $\|\mathbf{Ax} - \mathbf{b}\|$ which is the same as minimizing its square $\|\mathbf{Ax} - \mathbf{b}\|^2$

<u>**Gameplan**</u>:

- expand out the squared residual,
- take its first derivative, and
- set the derivative to zero

Linear Least Squares

Gameplan:

- expand out the squared residual,
- take its first derivative, and
- set the derivative to zero

Here, we can leverage the fact that the **squared** length of a vector is just its inner product with itself

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 = \langle \mathbf{A}\mathbf{x} - \mathbf{b}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle$$

Linear Least Squares

Gameplan:

- expand out the squared residual,
- take its first derivative, and
- set the derivative to zero

$$
\begin{aligned}
\|\mathbf{Ax} - \mathbf{b}\|^2 &= \langle \mathbf{Ax} - \mathbf{b}, \mathbf{Ax} - \mathbf{b} \rangle \\
&= \langle \mathbf{Ax}, \mathbf{Ax} \rangle - \langle \mathbf{Ax}, \mathbf{b} \rangle - \langle \mathbf{b}, \mathbf{Ax} \rangle + \langle \mathbf{b}, \mathbf{b} \rangle \\
&= \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} - 2\mathbf{b}^T \mathbf{Ax} + \|\mathbf{b}\|_2^2
\end{aligned}
$$

Linear Least Squares

Gameplan:

- expand out the squared residual,
- take its first derivative, and
- set the derivative to zero

$$\frac{\partial}{\partial \mathbf{x}}\left[\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{b}^T\mathbf{A}\mathbf{x} + \|\mathbf{b}\|_2^2\right] = 0$$

After dusting off your vector calculus identities, you'll obtain…

Linear Least Squares

$$\frac{\partial}{\partial \mathbf{x}}\left[\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{b}^T\mathbf{A}\mathbf{x}\right] = 0$$

then $$2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b} = 0$$

$$\boxed{\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}}$$

Normal equations

Linear Least Squares – normal eqns.

$$\begin{bmatrix} \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \end{bmatrix}$$

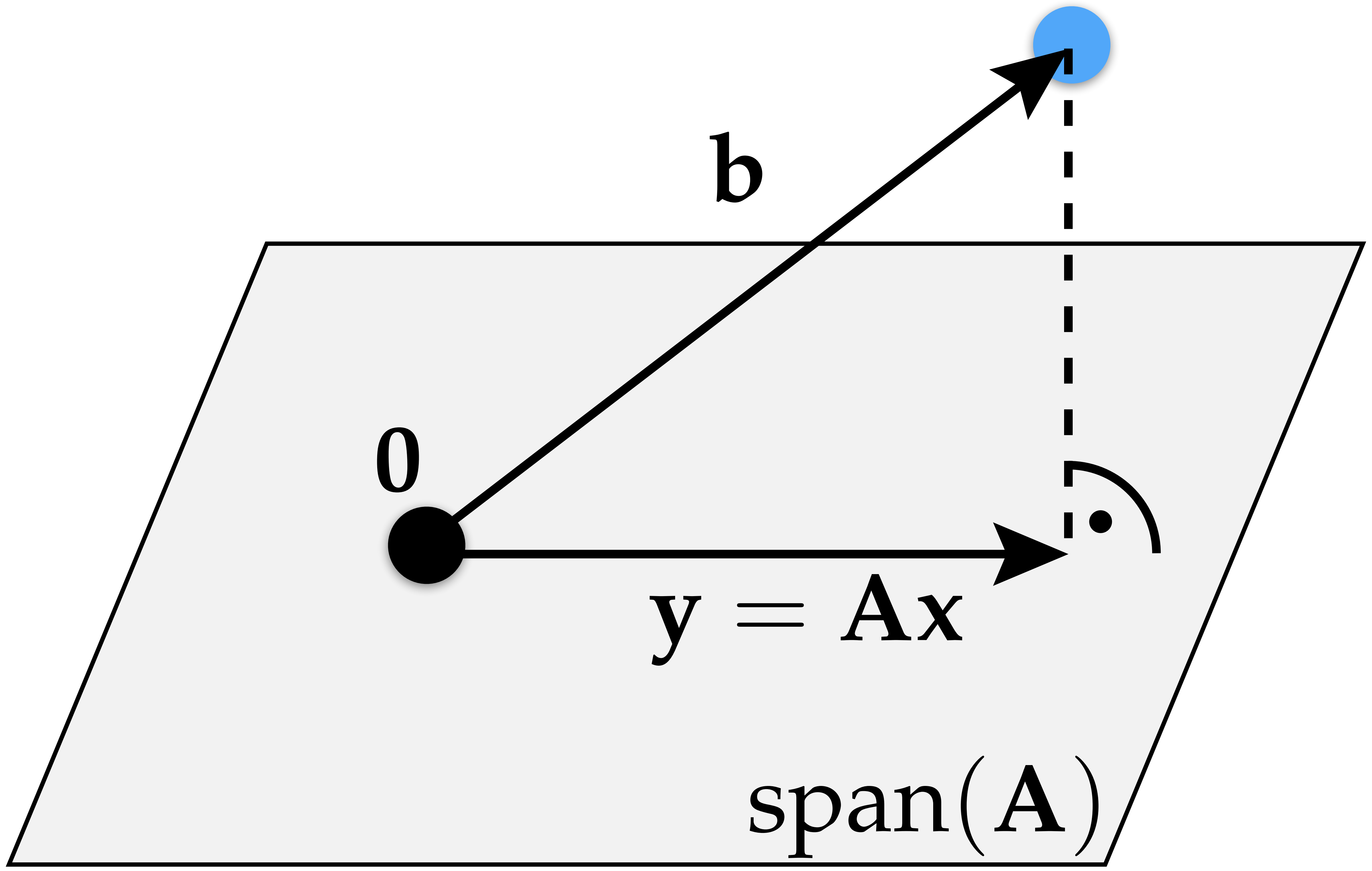
Linear Least Squares – normal eqns.

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$$

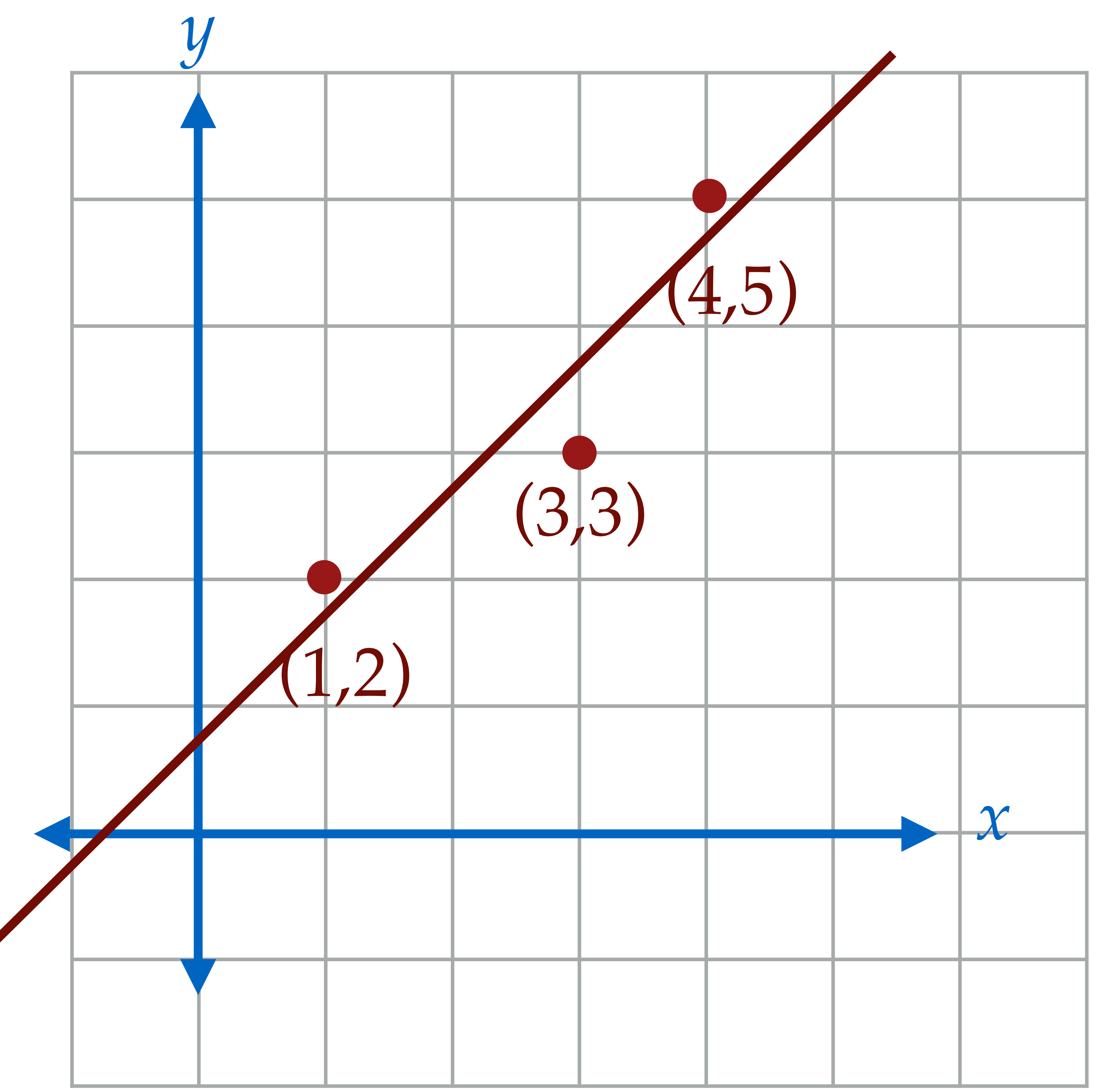
Linear Least Squares – normal eqns.

$$\begin{bmatrix} \mathbf{A}^T\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^T\mathbf{b} \end{bmatrix}$$

Why the "Normal" Equations?



Linear Least Squares – normal eqns.



$$m = ? \quad b = ?$$

$$\begin{pmatrix}1\\4\\3\end{pmatrix} m + \begin{pmatrix}1\\1\\1\end{pmatrix} b \neq \begin{pmatrix}2\\5\\3\end{pmatrix}$$

$$\vec{x}_1 m + \vec{x}_2 b \neq \vec{y}$$

$$\begin{pmatrix}1 & 1\\4 & 1\\3 & 1\end{pmatrix} \begin{pmatrix}m\\b\end{pmatrix} \neq \begin{pmatrix}2\\5\\3\end{pmatrix}$$

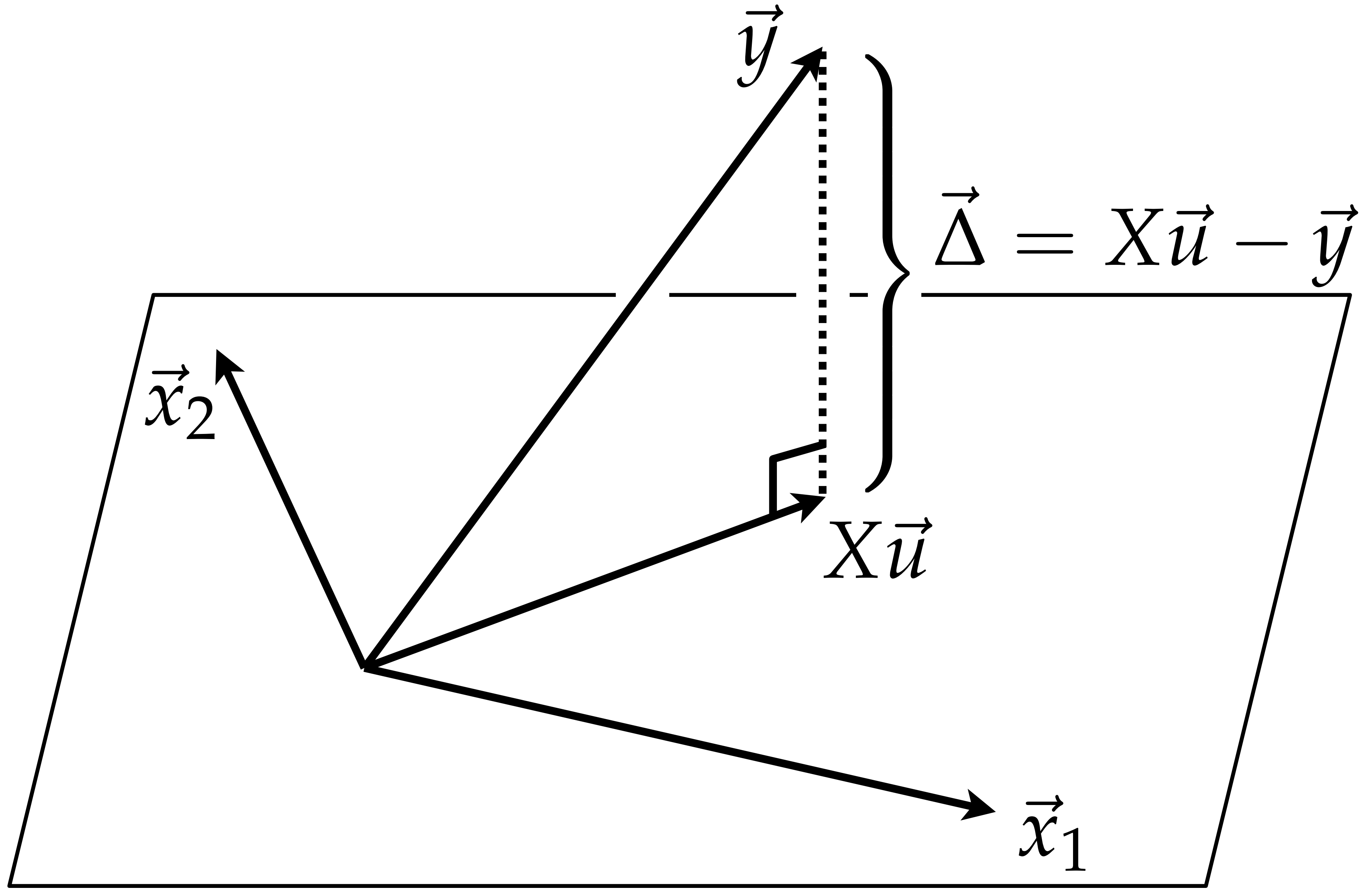
$$X\vec{u} \neq \vec{y}$$

$$X\vec{u} - \vec{y} = \vec{\Delta}$$

$$\min_{\vec{u}} \|X\vec{u} - \vec{y}\|^2$$

$$(X^\mathsf{T} X)\vec{u} = X^\mathsf{T}\vec{y}$$

$$\vec{u} = (X^\mathsf{T} X)^{-1} X^\mathsf{T}\vec{y}$$



(e.g., 2D plane in 3D)

Normal Equations – Conditioning

The normal equations *mathematically* model a least-squares solution to an overdetermined linear system

- *numerically* solving the system with stability is another issue…

In the fully-constrained (*i.e.*, square) case, we used the condition number to measure the stability of our solution

- we can proceed similarly with the fully-constrained system obtained by the orthogonal residual constraints added from the normal equations to our original overdetermined system

Normal Equations – Conditioning

Unfortunately, we can quickly run into stability issues when solving a system formed from the normal equations:

$$\mathrm{cond}(\mathbf{A}^T\mathbf{A}) \approx \mathrm{cond}(\mathbf{A})^2$$

Ouch! ☹

We'll look at how to solve such a system more robustly, e.g., without ever explicitly forming $\mathbf{A}^T\mathbf{A}$

Supplemental Readings

[Ascher & Greif] Section 5.8, 6.1

Errors and condition number, least squares and normal equations

[Heath] Sections 2.2, 2.3, 3.1 – 3.3

Uniqueness and conditioning, least squares and normal equations

[Solomon] Chapter 3

Designing and analyzing linear systems

Weighted Least Squares

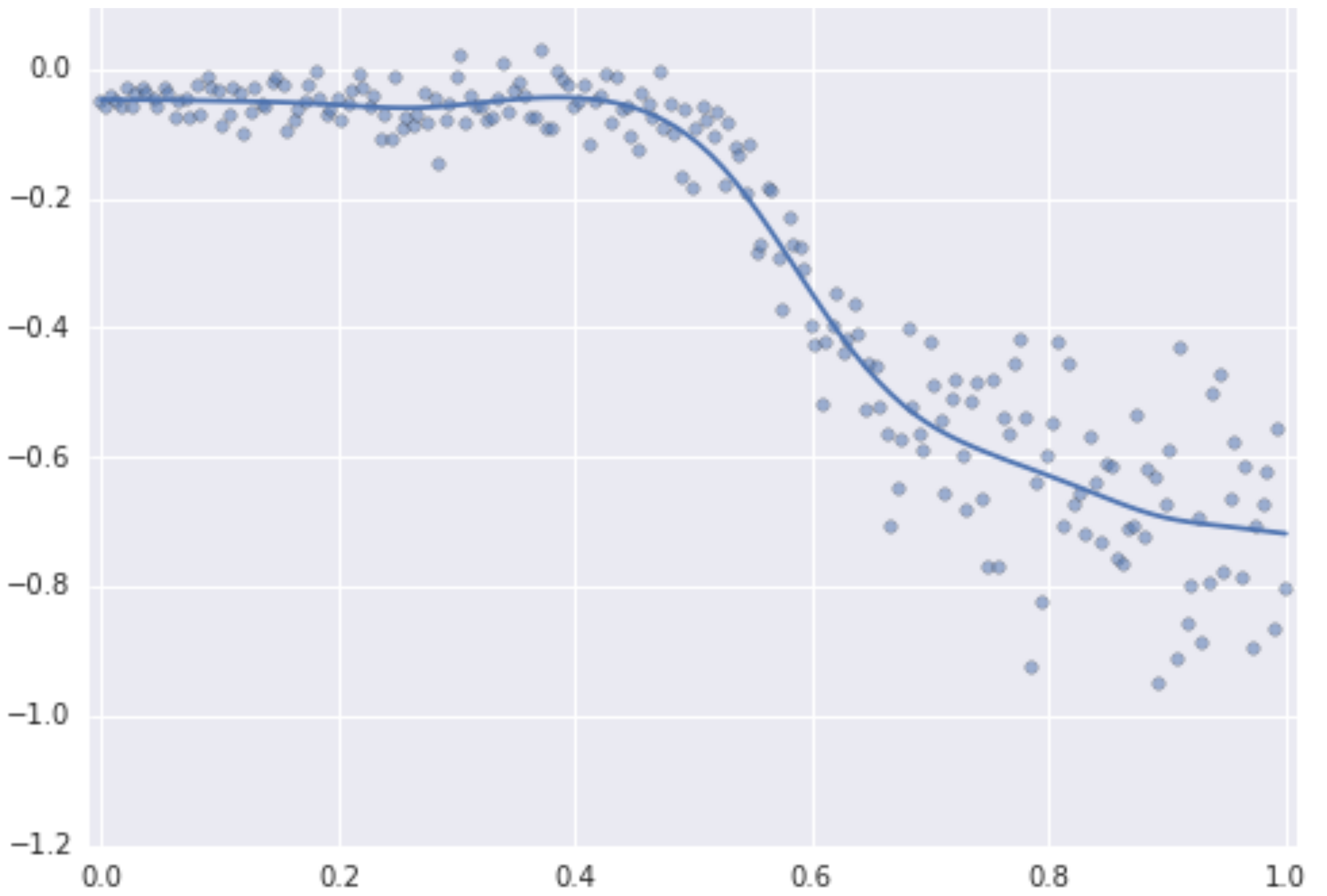
Linear Regression with Least Squares

How to fit simple (e.g. linear) model of n parameters to $m > n$ data points?

- least squares formulation

Can we treat data corruption?

- do we have extra information, e.g.:
  - relative importance of the data points
  - explicit model of the noise/corruption



$$\mathbf{Ax} \approx \mathbf{b}$$

$$\left[\mathbf{A}^T\mathbf{A}\right]\left[\mathbf{x}\right] = \left[\mathbf{A}^T\mathbf{b}\right]$$

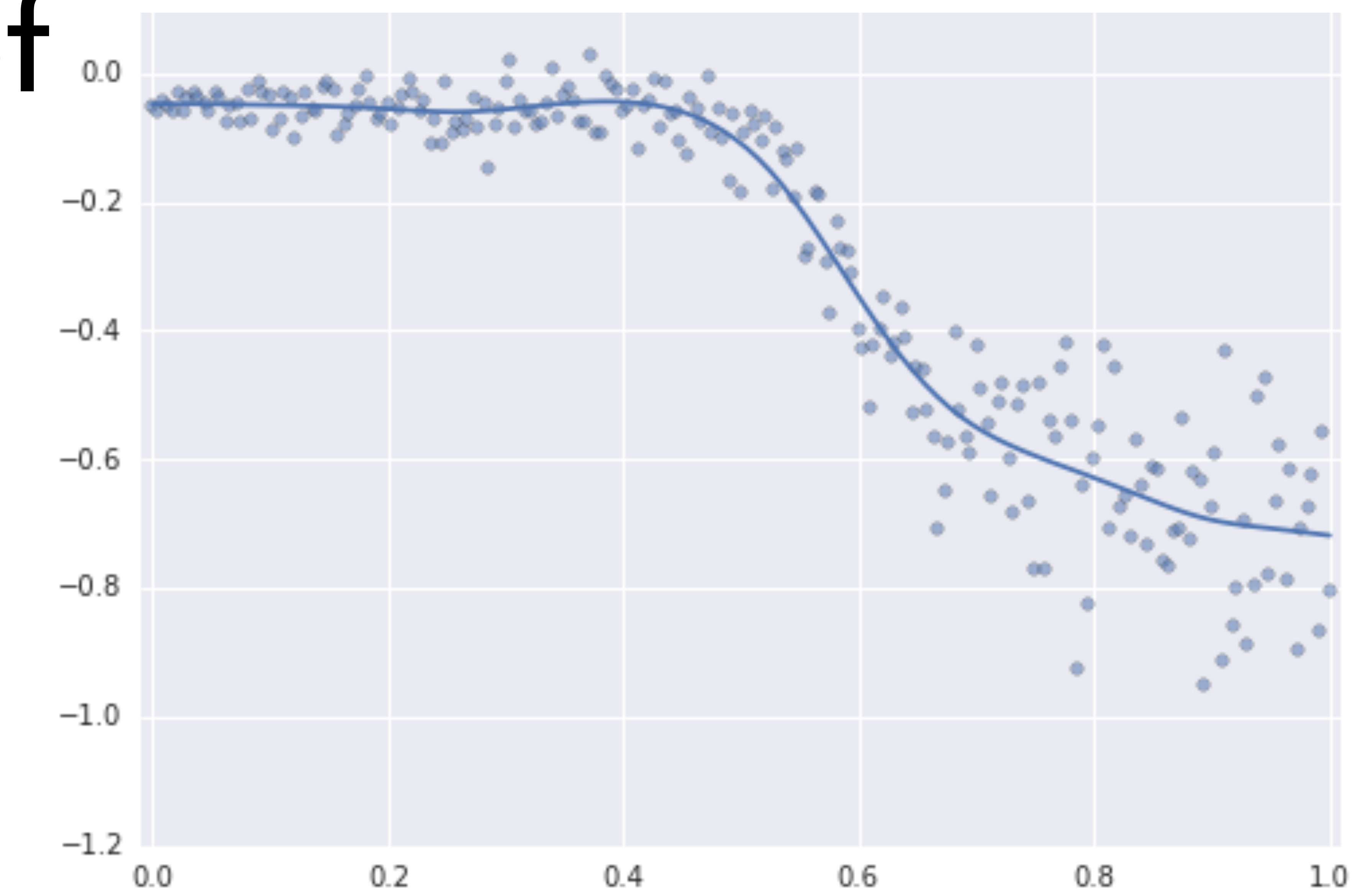
Linear Regression with Least Squares

How to fit simple (e.g. linear) model of n parameters to $m > n$ data points?

- least squares formulation

Can we treat data corruption?

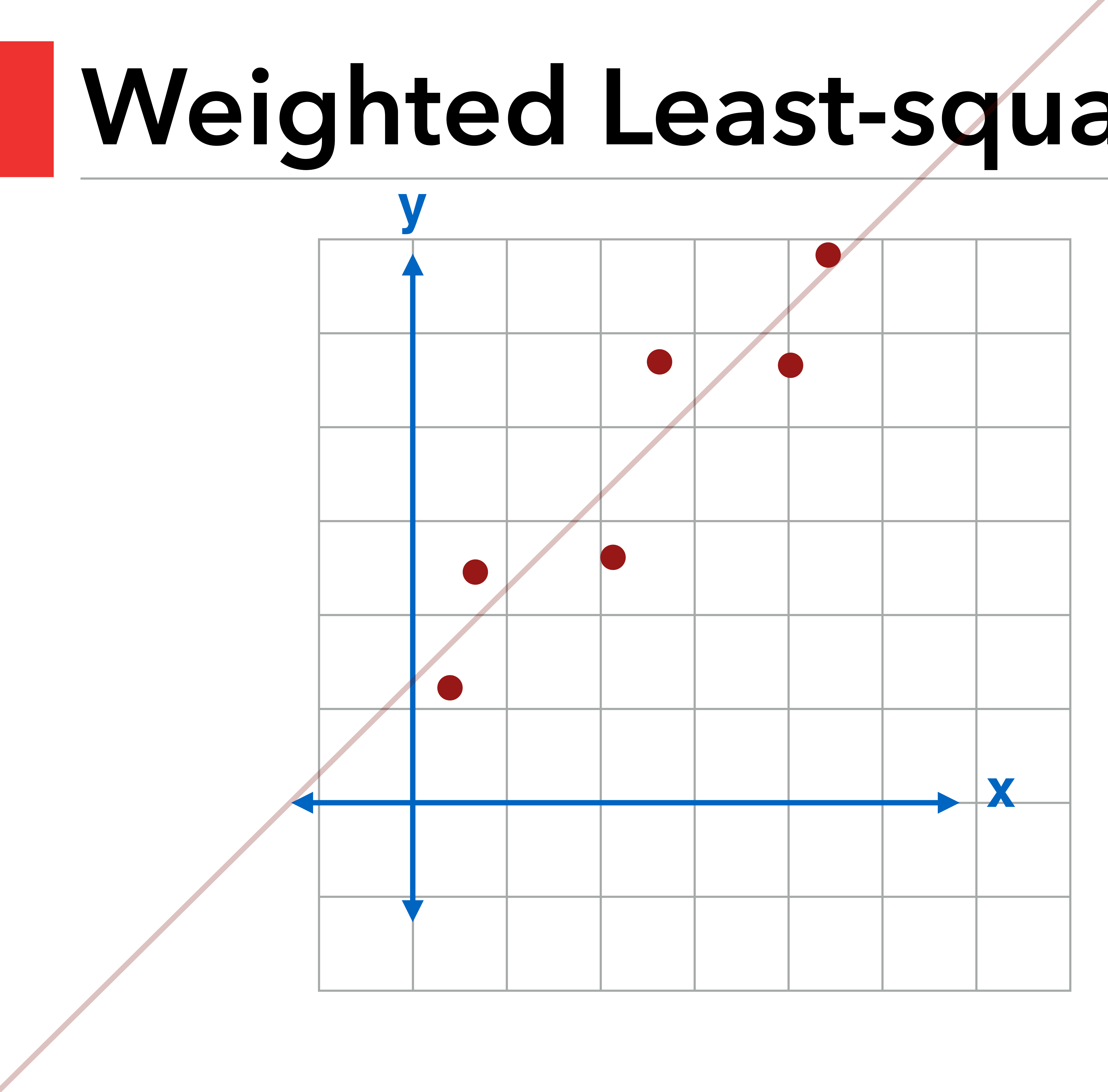
- do we have extra information, e.g.:
  - relative importance of the data points
  - explicit model of the noise/corruption



$$\mathbf{A}\mathbf{x} \approx \mathbf{b}$$

$$\left[\mathbf{A}^T\mathbf{A}\right]\left[\mathbf{x}\right] = \left[\mathbf{A}^T\mathbf{b}\right]$$

Weighted Least-squares Estimation



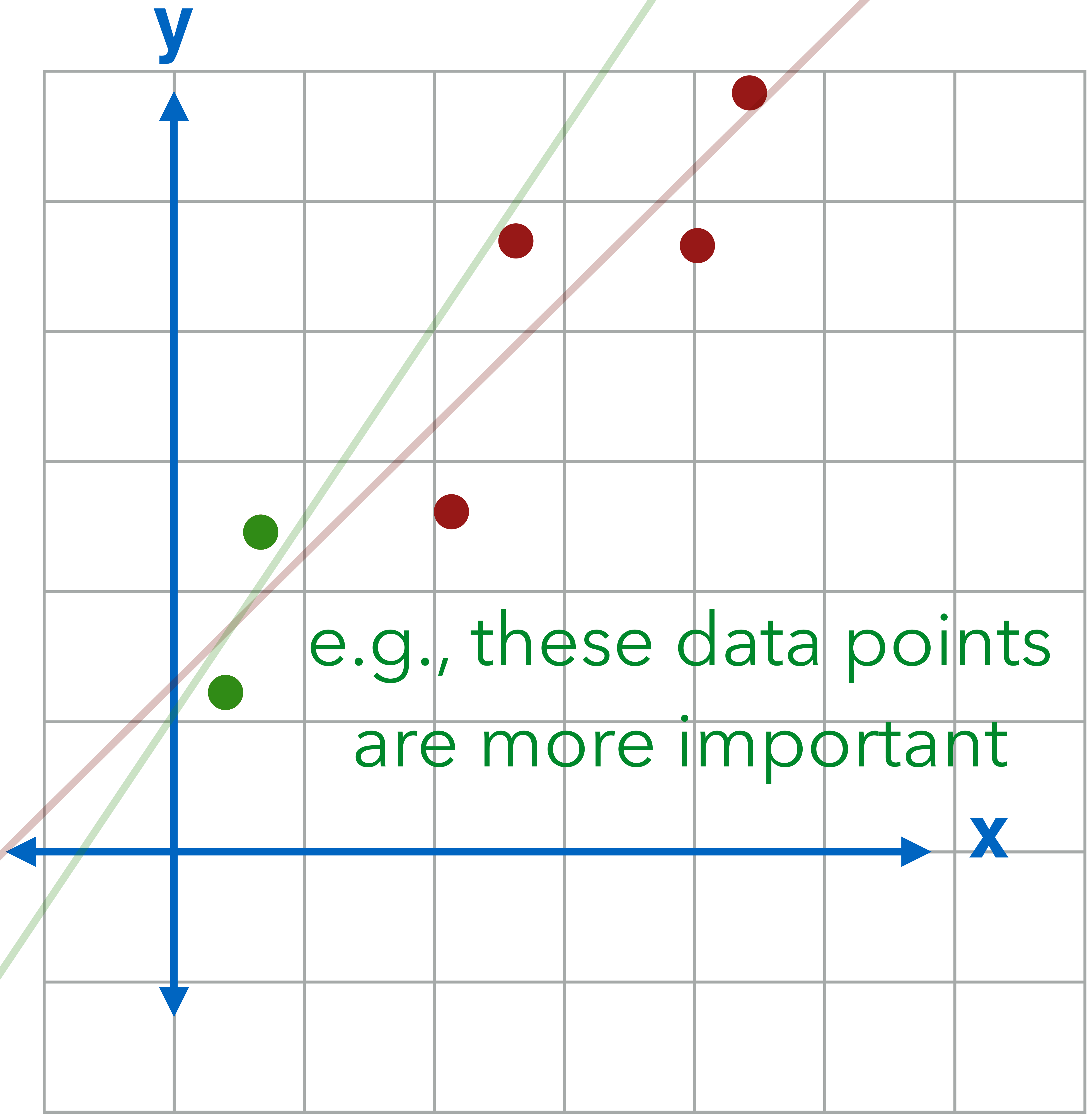
$$y_i = mx_i + b$$

minimize:

$\sum_{i=1}^{n} [(mx_i + b) - y_i]^2$

Standard least-squares treats every point as being *equally important*

Weighted Least-squares Estimation



$$y_i = mx_i + b$$

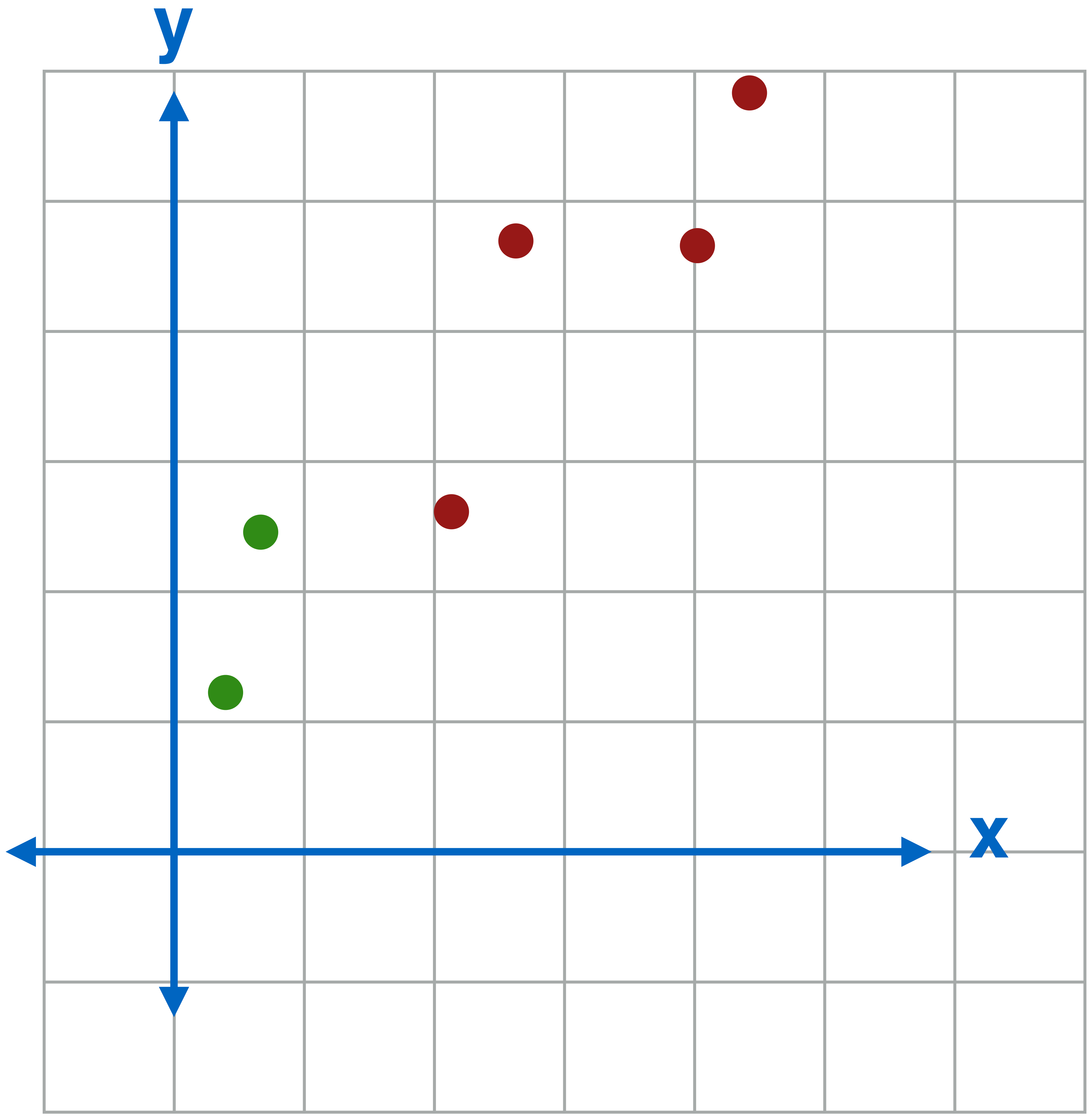
minimize:

$\sum_{i=1}^{n}[(mx_i + b) - y_i]^2$

$\sum_{i=1}^{n}[\underbrace{w_i}_{\text{weight}}((mx_i + b) - y_i)]^2$

Weighted least-squares allows us to attribute more importance to some points, relative to others

Weighted Least-squares Estimation



$$y_i = mx_i + b$$

minimize:

$$\sum_{i=1}^{n}[(mx_i + b) - y_i]^2$$

$$\sum_{i=1}^{n}[w_i((mx_i + b) - y_i)]^2$$

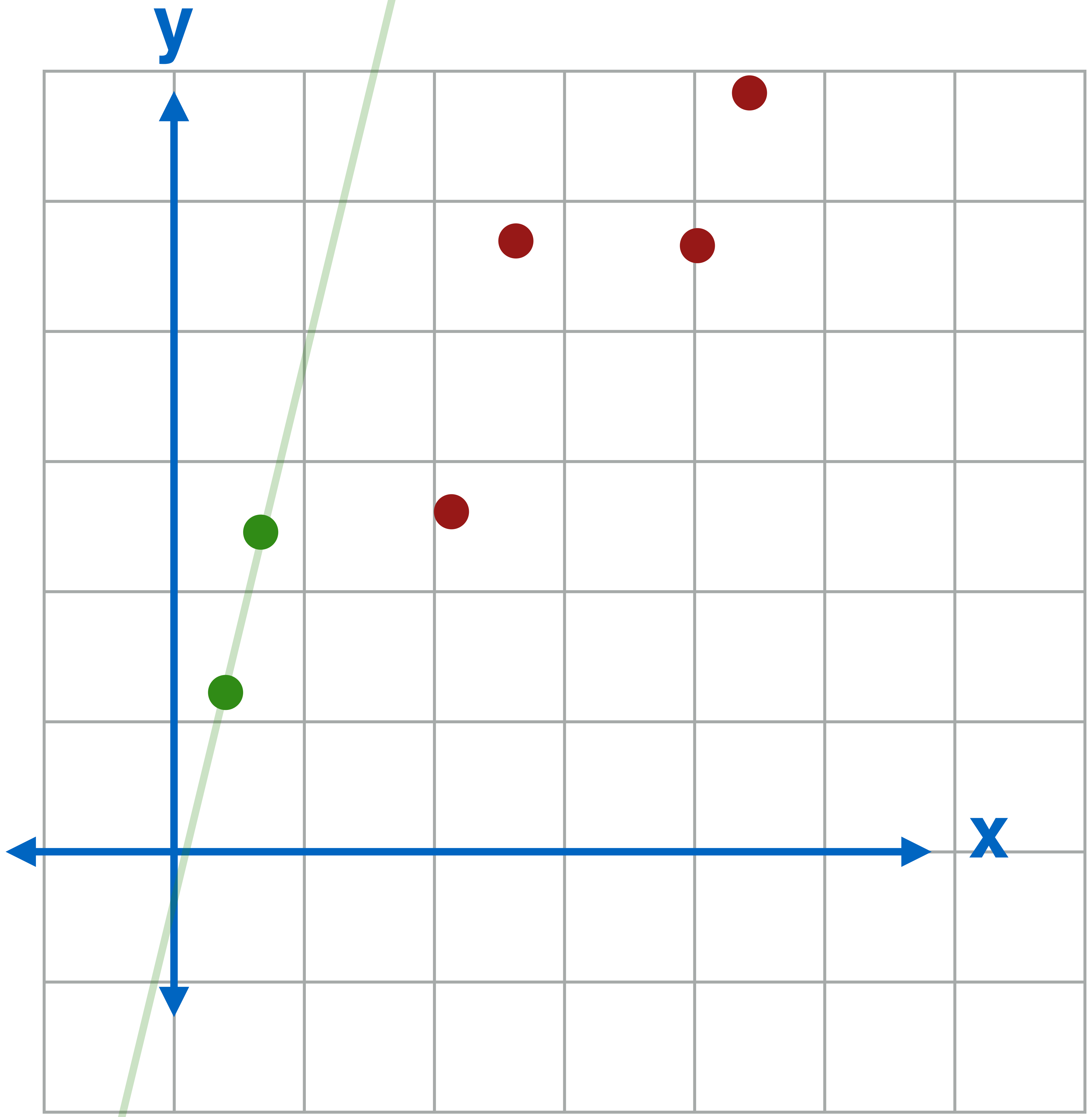
$$\sum_{i=1}^{n}[w_i \Delta_i]^2$$

$$[w_1\Delta_1]^2 + [w_2\Delta_2]^2 + [w_3\Delta_3]^2 + [w_4\Delta_4]^2 + [w_5\Delta_5]^2 + [w_6\Delta_6]^2$$

$$w_1 = w_2 = 1$$

$$w_3 = \ldots = w_6 = 0$$

Weighted Least-squares Estimation



$$y_i = mx_i + b$$

minimize:

$$\sum_{i=1}^{n}[(mx_i + b) - y_i]^2$$

$$\sum_{i=1}^{n}[w_i((mx_i + b) - y_i)]^2$$

$$\sum_{i=1}^{n}[w_i\Delta_i]^2$$

$$[w_1\Delta_1]^2 + [w_2\Delta_2]^2 + [w_3\Delta_3]^2 + [w_4\Delta_4]^2 + [w_5\Delta_5]^2 + [w_6\Delta_6]^2$$

$$w_1 = w_2 = 1$$

$$w_3 = \ldots = w_6 = 0$$

- we can recover fully-constrained fits as a special instance of WLS

From LS to Weighted LS

Consider the least-squares fit of a line (i.e., two parameters) to $n > 2$ points

- this fit minimizes the squared difference between model evaluations and observations:

$$\min_{m,b} \sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

From LS to Weighted LS

We know how to move from an expression of the residual...

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$X\vec{u} - \vec{y} = \vec{\Delta}$$

$$\min_{m,b} \sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

From LS to Weighted LS

We know how to move from an expression of the residual, to the squared error (which we seek to minimize)

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$X\vec{u} - \vec{y} = \vec{\Delta}$$

$$E(\vec{u}) = \|X\vec{u} - \vec{y}\|^2$$

$$\min_{m,b} \sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

From LS to Weighted LS

What remains unclear is the impact that weighting data points (non-uniformly) will have on our original LS formulation (and solution)

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$X\vec{u} - \vec{y} = \vec{\Delta}$$

$$E(\vec{u}) = \|X\vec{u} - \vec{y}\|^2$$

$$\min_{m,b} \sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i((mx_i + b) - y_i)]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i \Delta_i]^2$$

From LS to Weighted LS

To start, we can express the weighted least-squares residual/error in matrix-vector form

- the *weight matrix* is a diagonal matrix of the per data-point weights

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

$$W = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix}$$

$$\min_{m,b} \textstyle\sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

$$\min_{m,b} \textstyle\sum_{i=1}^{n} [w_i((mx_i + b) - y_i)]^2$$

$$\min_{m,b} \textstyle\sum_{i=1}^{n} [w_i \Delta_i]^2$$

From LS to Weighted LS

To start, we can express the weighted least-squares residual/error in matrix-vector form

- the *weight matrix* is a diagonal matrix of the per data-point weights

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

$$\begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_n \end{pmatrix}$$

$$\min_{m,b} \sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i((mx_i + b) - y_i)]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i \Delta_i]^2$$

From LS to Weighted LS

To start, we can express the weighted least-squares residual/error in matrix-vector form

- the *weight matrix* is a diagonal matrix of the per data-point weights

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

$$\begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_n \end{pmatrix}$$

$$\min_{m,b} \sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i((mx_i + b) - y_i)]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i \Delta_i]^2$$

From LS to Weighted LS

To start, we can express the weighted least-squares residual/error in matrix-vector form

- the *weight matrix* is a diagonal matrix of the per data-point weights

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

$$\begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_n \end{pmatrix}$$

$$\min_{m,b} \sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i((mx_i + b) - y_i)]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i \Delta_i]^2$$

From LS to Weighted LS

To start, we can express the weighted least-squares residual/error in matrix-vector form

- the *weight matrix* is a diagonal matrix of the per data-point weights

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

$$\begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_n \end{pmatrix}$$

$$\min_{m,b} \sum_{i=1}^{n} [(mx_i + b) - y_i]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i((mx_i + b) - y_i)]^2$$

$$\min_{m,b} \sum_{i=1}^{n} [w_i \Delta_i]^2$$

From LS to Weighted LS

Let's proceed as we did for LS to minimize the weighted squared residual

- take derivative,
- set to 0, and
- try to express a solution to the resulting system

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

$$E(\vec{u}) = \|WX\vec{u} - W\vec{y}\|^2$$

From LS to Weighted LS

Let's proceed as we did for LS to minimize the weighted squared residual

- take derivative,
- set to 0, and
- try to express a solution to the resulting system

$$E(\vec{u}) = \|WX\vec{u} - W\vec{y}\|^2$$

$$\frac{dE}{d\vec{u}} = 2X^TW^T(WX\vec{u} - W\vec{y}) = 0$$

From LS to Weighted LS

Let's proceed as we did for LS to minimize the weighted squared residual

- take derivative,
- set to 0, and
- try to express a solution to the resulting system

$$E(\vec{u}) = \|WX\vec{u} - W\vec{y}\|^2$$

$$\frac{dE}{d\vec{u}} = 2X^TW^T(WX\vec{u} - W\vec{y}) = 0$$

From LS to Weighted LS

Let's proceed as we did for LS to minimize the weighted squared residual

- take derivative,
- set to 0, and
- try to express a solution to the resulting system

$$E(\vec{u}) = \|WX\vec{u} - W\vec{y}\|^2$$

$$\frac{dE}{d\vec{u}} = 2X^TW^T(WX\vec{u} - W\vec{y}) = 0$$

From LS to Weighted LS

$$(AB)^T = (B^T A^T)$$

Let's proceed as we did for LS to minimize the weighted squared residual

- take derivative,
- set to 0, and
- try to express a solution to the resulting system

$$E(\vec{u}) = \|WX\vec{u} - W\vec{y}\|^2$$

$$\frac{dE}{d\vec{u}} = 2X^T W^T (WX\vec{u} - W\vec{y}) = 0$$

From LS to Weighted LS

$$(AB)^T = (B^T A^T)$$

Let's proceed as we did for LS to minimize the weighted squared residual

$$E(\vec{u}) = \|WX\vec{u} - W\vec{y}\|^2$$

$$\frac{dE}{d\vec{u}} = 2X^TW^T(WX\vec{u} - W\vec{y}) = 0$$

for m unknowns with $n > m$ constraints:

- take derivative,
- set to 0, and
- try to express a solution to the resulting system

$$\underbrace{X^T \overbrace{W^T W}^{n\times n \;\; n\times n} X \vec{u}}_{\substack{m\times n \;\; [n\times n] \;\; n\times m \\ \overline{m\times 1}}} = \underbrace{\overbrace{X^T W^T W}^{m\times n \;\; [n\times n] \;\; n\times 1}\vec{y}}_{m\times 1}$$

From LS to Weighted LS

$$(AB)^T = (B^T A^T)$$

Let's proceed as we did for LS to minimize the weighted squared residual

- take derivative,
- set to 0, and
- try to express a solution to the resulting system

$$E(\vec{u}) = \|WX\vec{u} - W\vec{y}\|^2$$

$$\frac{dE}{d\vec{u}} = 2X^T W^T (WX\vec{u} - W\vec{y}) = 0$$

$$X^T W^T W X \vec{u} = X^T W^T W \vec{y}$$

$$\vec{u} = \underbrace{(X^T W^T W X)}_{m \times m}{}^{-1} \underbrace{X^T W^T W \vec{y}}_{m \times 1}$$

From LS to Weighted LS

$$(AB)^T = (B^T A^T)$$

Let's proceed as we did for LS to minimize the weighted squared residual

- take derivative,
- set to 0, and
- try to express a solution to the resulting system

$$E(\vec{u}) = \|WX\vec{u} - W\vec{y}\|^2$$

$$\frac{dE}{d\vec{u}} = 2X^TW^T(WX\vec{u} - W\vec{y}) = 0$$

$$X^TW^TWX\vec{u} = X^TW^TW\vec{y}$$

$$\vec{u} = (X^TW^TWX)^{-1}X^TW^TW\vec{y}$$

$$\vec{u} = (X^TW^2X)^{-1}X^TW^2\vec{y}$$

From LS to Weighted LS

$$(AB)^T = (B^T A^T)$$

We arrive at a modified set of the normal equations that take the constraint weights into account

- setting $w_i = 1, \forall i$ yields our original LS formulation and its solution formulation (i.e., the normal equations)

$$\vec{u} = (X^T W^2 X)^{-1} X^T W^2 \vec{y}$$

$$W^2 = \begin{pmatrix} w_1^2 & 0 & \dots & 0 \\ 0 & w_2^2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w_n^2 \end{pmatrix}$$

$$\vec{u} = (X^T X)^{-1} X^T \vec{y}$$

Weighted LS – Conditioning

Don't forget that – despite formulating an expression for the WLS solution – solving this numerically is a completely different issue

- earlier, we saw that the conditioning of the normal equations was $\text{cond}(\mathbf{A}^T\mathbf{A}) \approx \text{cond}(\mathbf{A})^2$ ☹
- what's the condition number of the WLS solution: $\mathbf{A}^T\mathbf{W}^2\mathbf{A}$?

$$\text{cond}(\mathbf{A}^T\mathbf{W}^2\mathbf{A}) \approx \text{cond}(\mathbf{A})^2$$

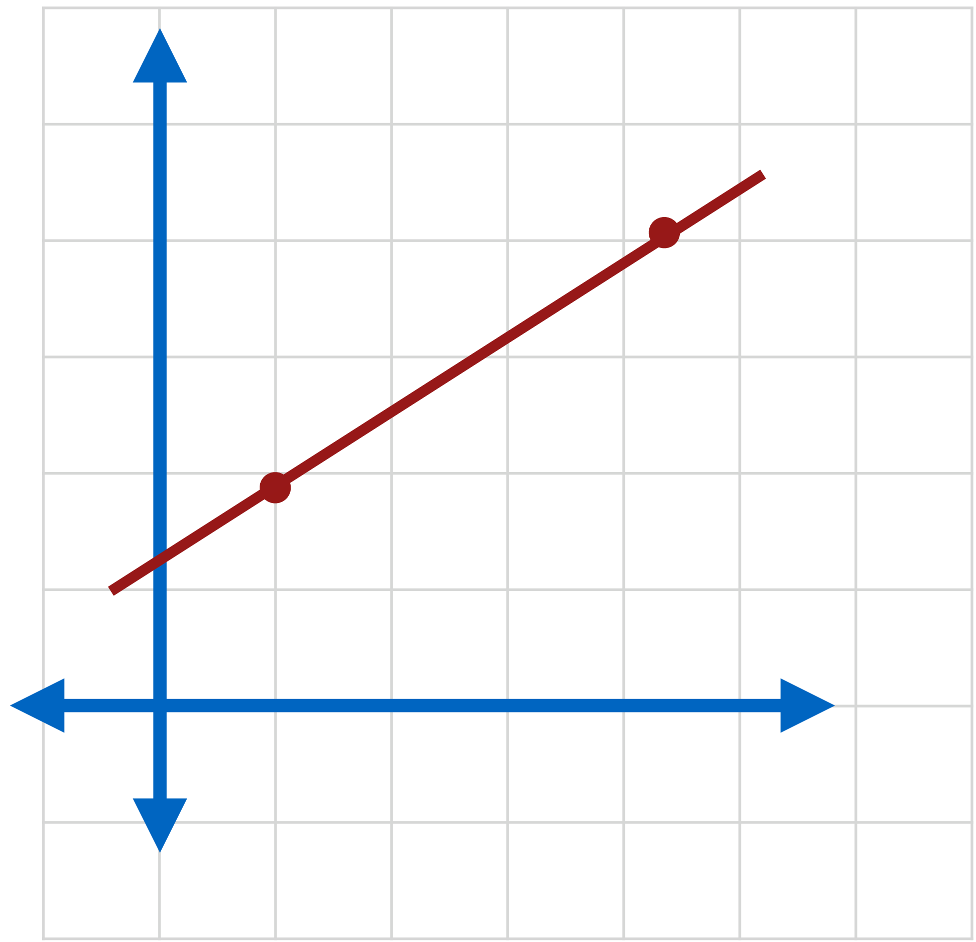
Summary – Solutions of Linear Systems

fully constrained linear equations

[matrix inverse]

$$\mathbf{Ax} = \mathbf{y} \ (\mathbf{A} \text{ is } n \times n)$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$



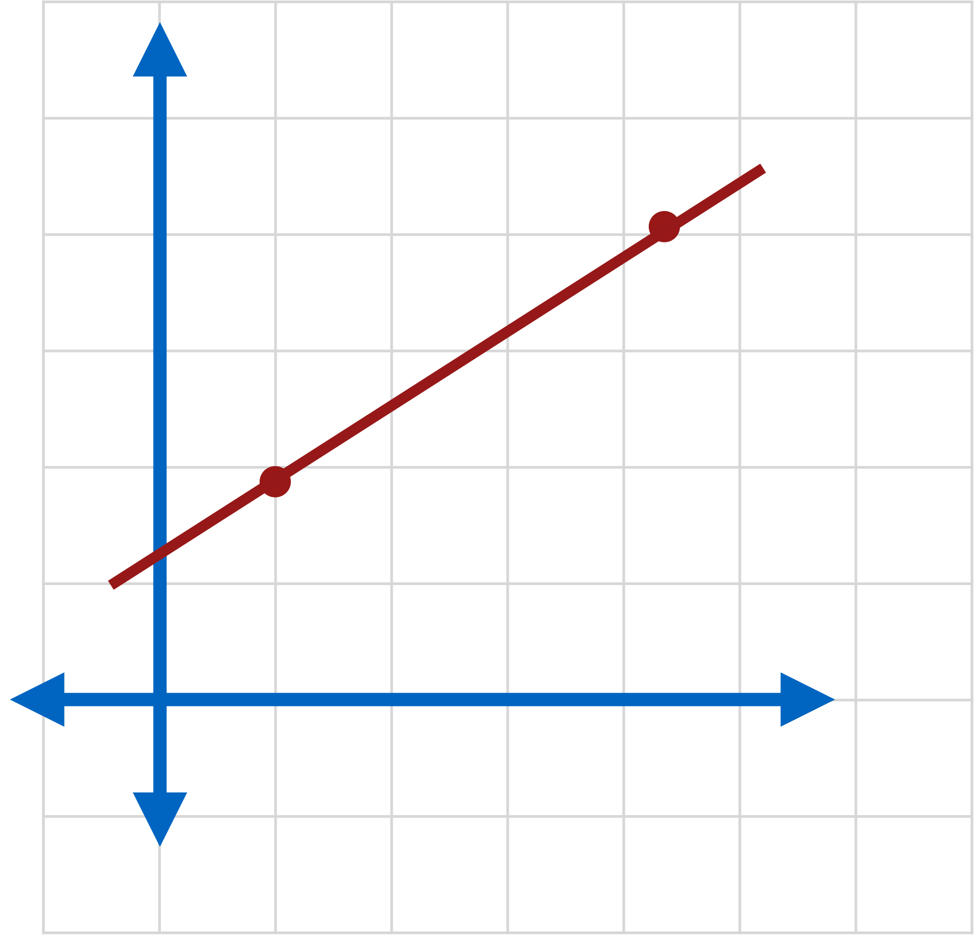
Summary – Solutions of Linear Systems

fully constrained linear equations

[matrix inverse]

$$\mathbf{A}\mathbf{x} = \mathbf{y} \ (\mathbf{A} \text{ is } n \times n)$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$



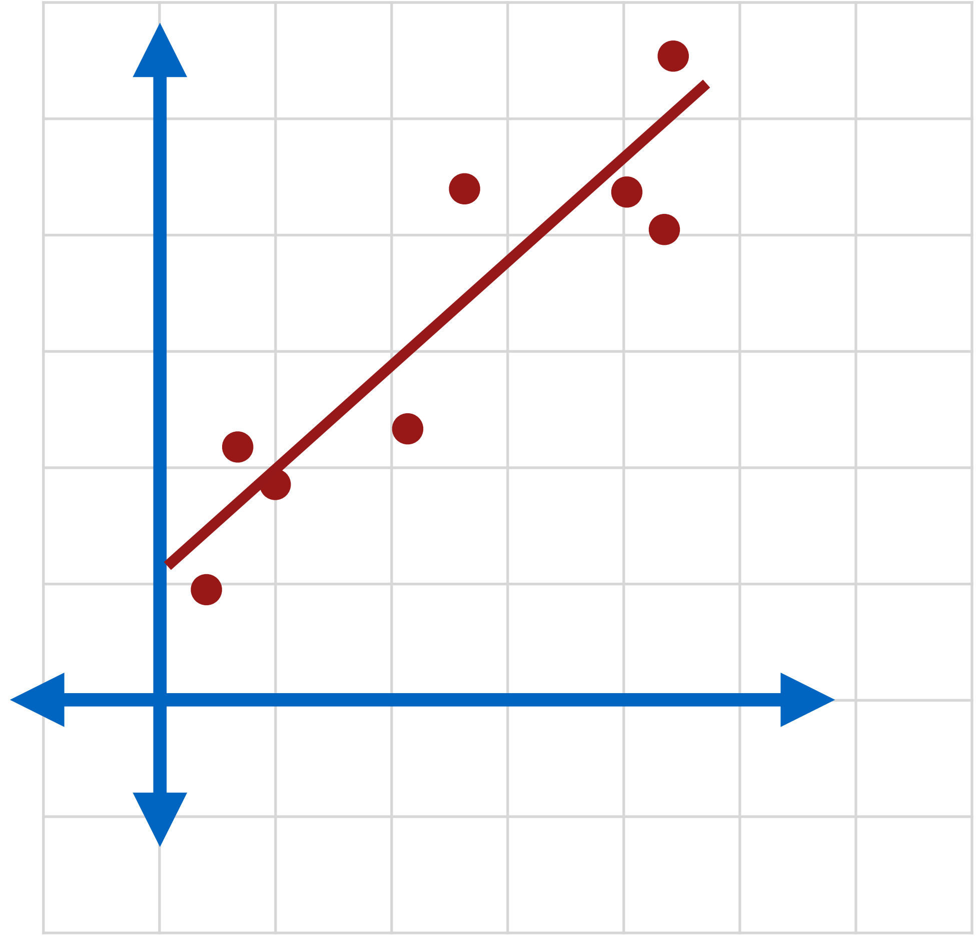
over-constrained linear equations

[least squares]

$$E(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2$$

$$\mathbf{A}\mathbf{x} = \mathbf{y} \ (\mathrm{A} \text{ is } n \times p,\ n \geq p)$$

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$$



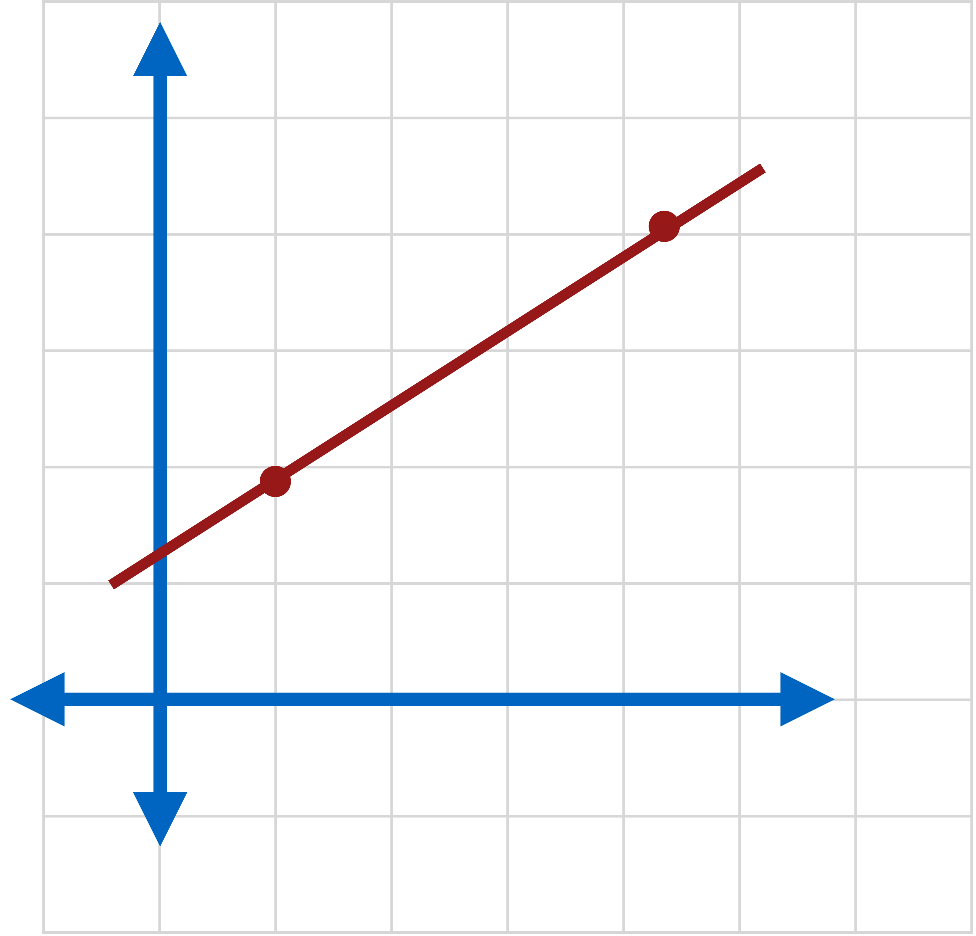
Summary – Solutions of Linear Systems

fully constrained linear equations

[matrix inverse]

$$\mathbf{Ax} = \mathbf{y} \ (\mathbf{A} \text{ is } n \times n)$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$



[least squares]

$$E(\mathbf{x}) = ||\mathbf{Ax} - \mathbf{y}||^2$$

$$\mathbf{Ax} = \mathbf{y} \ (\mathbf{A} \text{ is } n \times p, n \geq p)$$

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$$

over-constrained linear equations

[weighted least squares]

$$E(\mathbf{x}) = ||\mathbf{W}(\mathbf{Ax} - \mathbf{y})||^2 = ||\mathbf{WAx} - \mathbf{Wy}||^2$$

$$\mathbf{WAx} = \mathbf{Wy} \ (\mathbf{A} \text{ is } n \times p \ , n \geq p)$$

$$\mathbf{x} = (\mathbf{A}^T\mathbf{W}^2\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^2\mathbf{y}$$