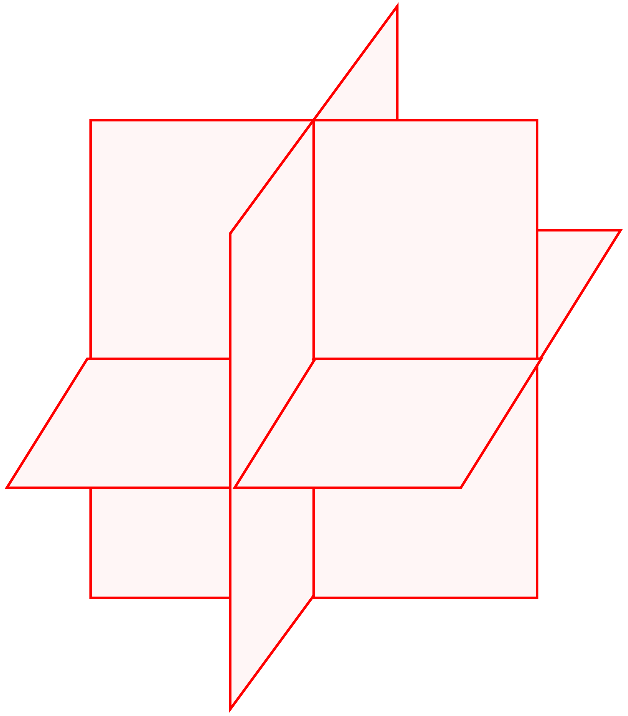
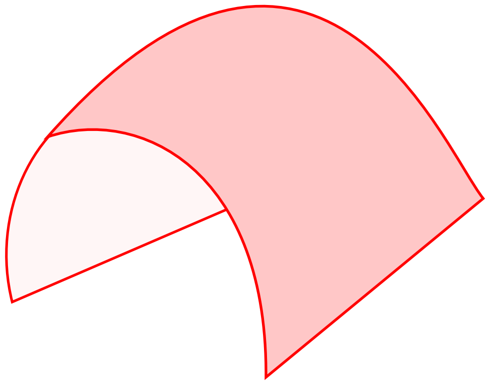


Some Recent Advances in Non-convex Optimization

Purushottam Kar

IIT KANPUR



Outline of the Talk

- Recap of Convex Optimization
- Why Non-convex Optimization?
- Non-convex Optimization: A Brief Introduction
- **Robust Regression**: A Non-convex Approach
- Robust Regression: Application to Face Recognition
- **Robust PCA**: A Sketch and Application to Foreground Extraction in Images

Recap of Convex Optimization

Convex Optimization

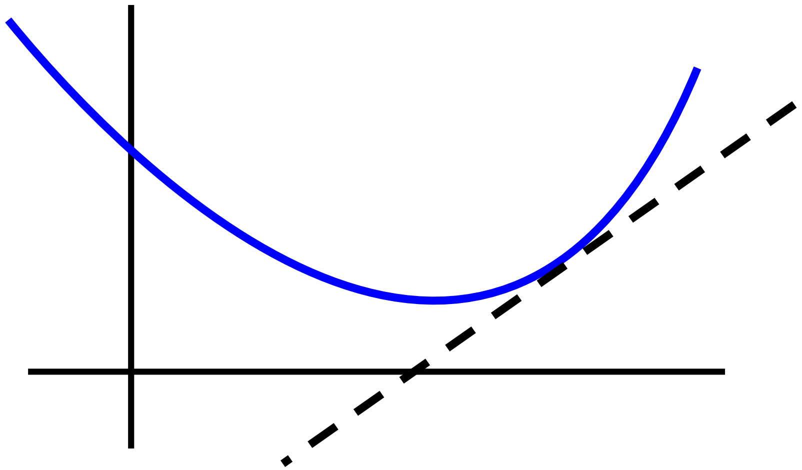
$$\min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x})$$

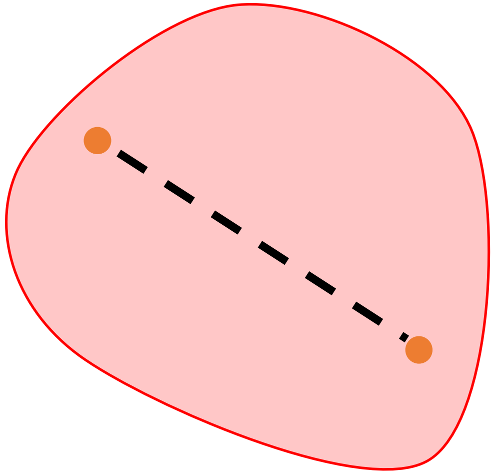
$f : \mathbb{R}^d \to \mathbb{R}$

Convex function

$\mathcal{C} \subseteq \mathbb{R}^d$

Convex set





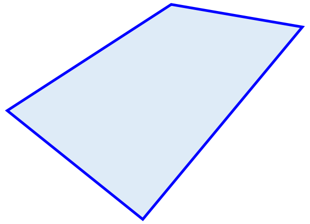
Examples

Linear Programming

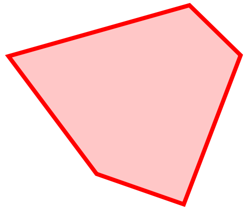
$$\min_{\mathbf{x}\in\mathbb{R}^d} \ \mathbf{a}^\top \mathbf{x}$$

$$s.t.\ \mathbf{b}_i^\top \mathbf{x} \le c_i$$

f



$\mathcal{C}$

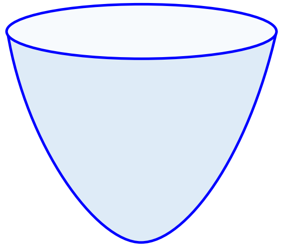


Quadratic Programming

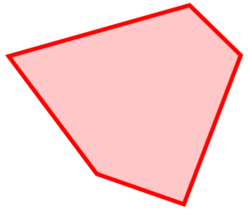
$$\min_{\mathbf{x}\in\mathbb{R}^d} \ \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{a}^\top \mathbf{x}$$

$$s.t.\ \mathbf{b}_i^\top \mathbf{x} \le c_i$$

f



$\mathcal{C}$

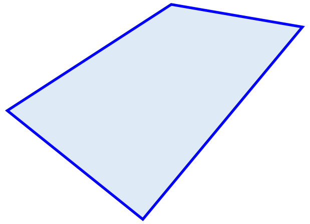


Semidefinite Programming

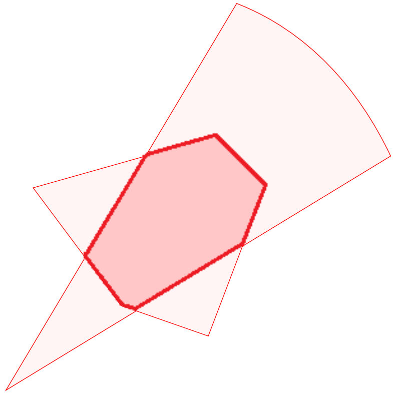
$$\min_{\mathbf{X}\succeq\mathbf{0}} \ \mathbf{A}^\top \mathbf{X}$$

$$s.t.\ \mathbf{B}_i^\top \mathbf{X} \le c_i$$

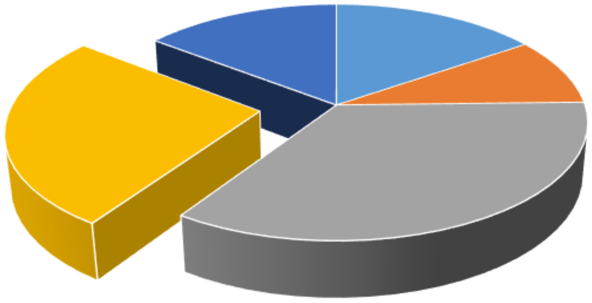
f



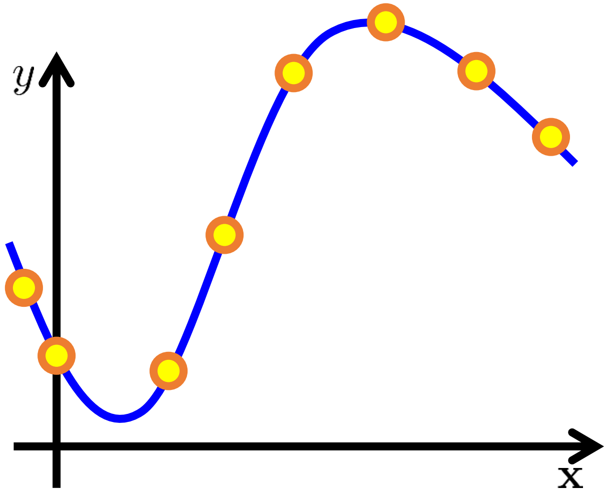
$\mathcal{C}$



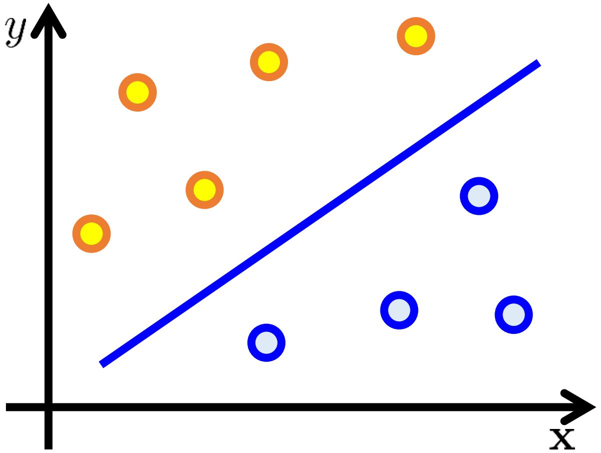
Applications



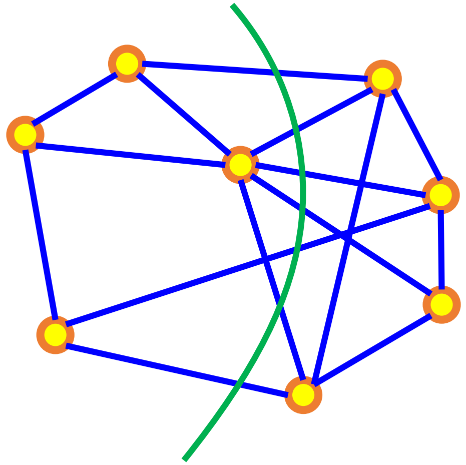
Resource Allocation



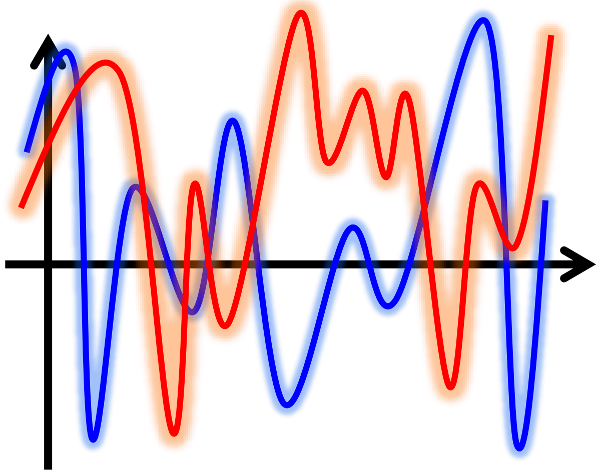
Regression



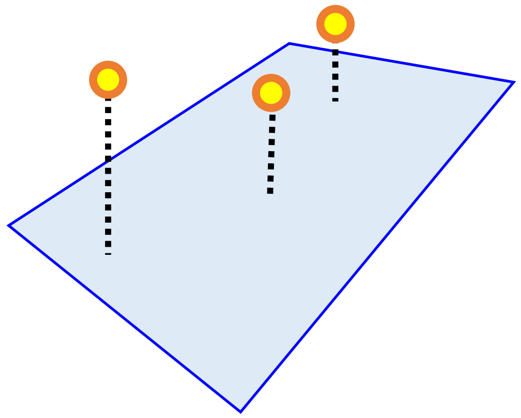
Classification



Clustering/Partitioning



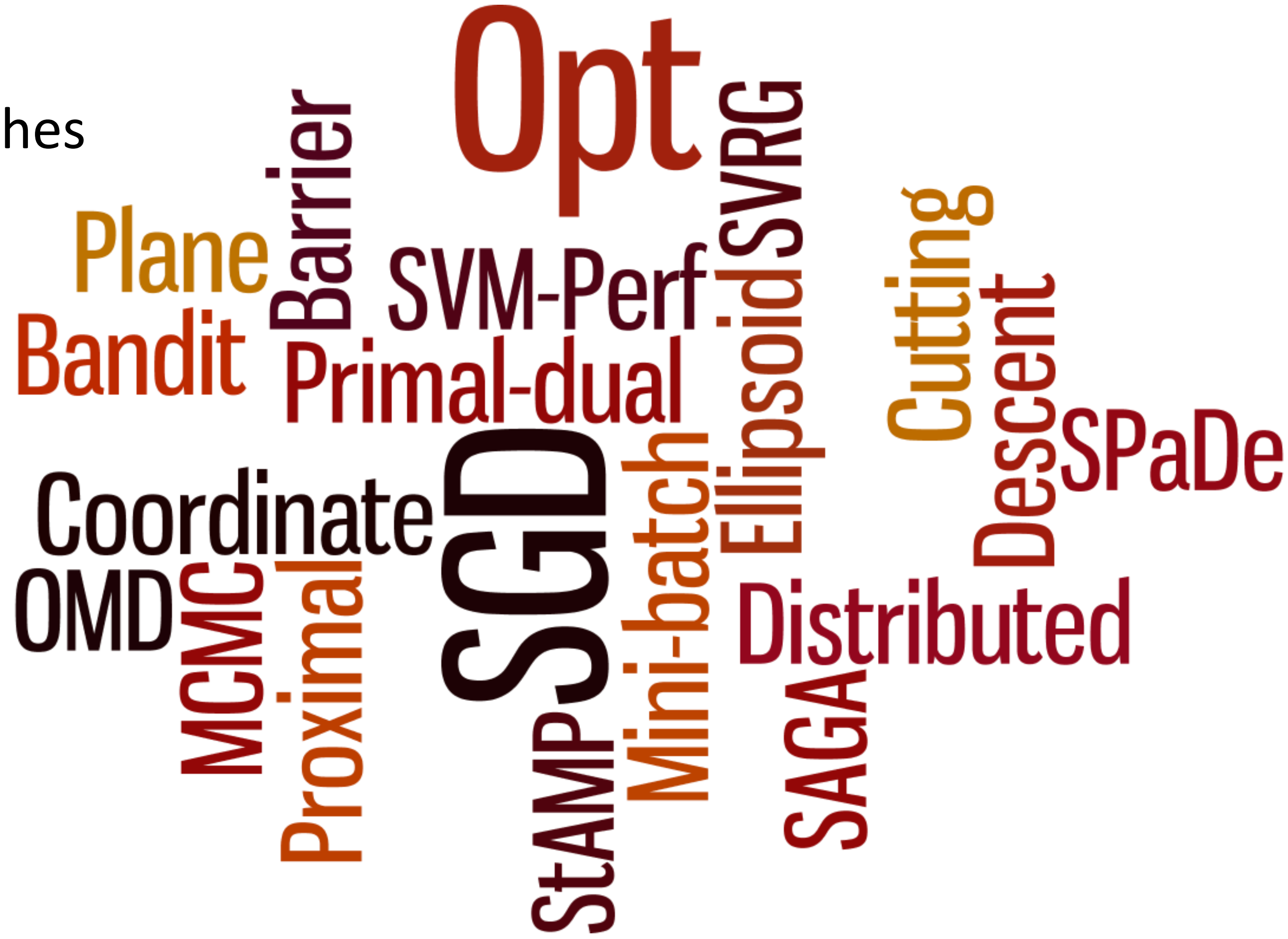
Signal Processing



Dimensionality Reduction

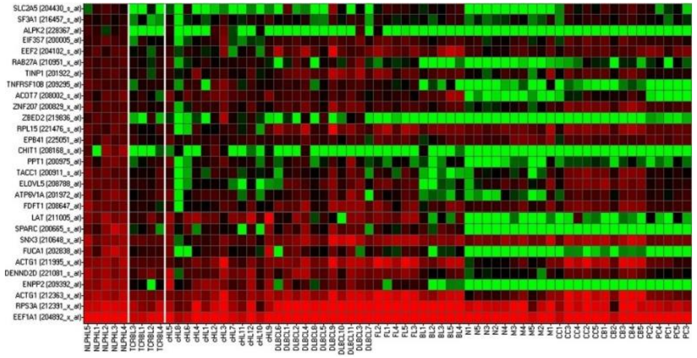
Techniques

- Projected (Sub)gradient Methods
  - Stochastic, mini-batch variants
  - Primal, dual, primal-dual approaches
  - Coordinate update techniques
- Interior Point Methods
  - Barrier methods
  - Annealing methods
- Other Methods
  - Cutting plane methods
  - Accelerated routines
  - Proximal methods
  - Distributed optimization
  - Derivative-free optimization



Why Non-convex Optimization?

Gene Expression Analysis



DNA micro-array gene expression data

… n subjects

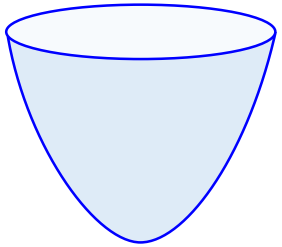
$${}_i = (\mathbf{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$$

Linear model: $y_i \approx \mathbf{x}_i^\top \mathbf{w}^*$

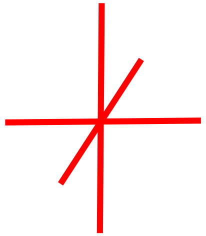
Challenge: $\mathbf{w}^*$ is sparse!

$$\min_{\mathbf{w}\in\mathcal{B}_0^p(s)} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w})^2$$

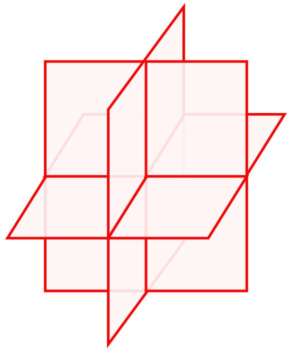
f



$\mathcal{C}$



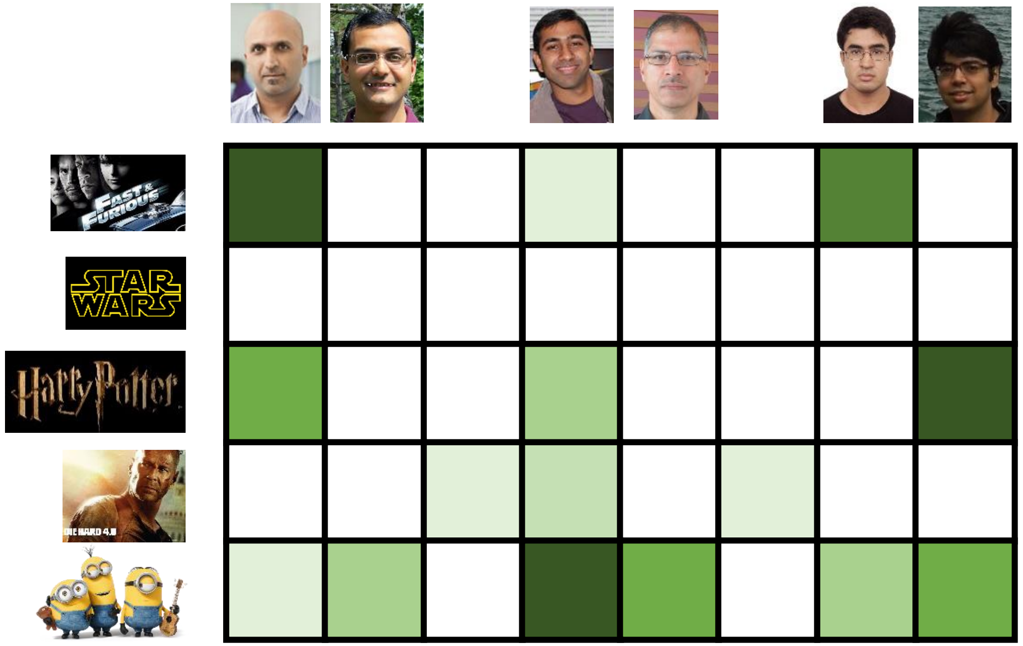
$p = 3, s = 1$

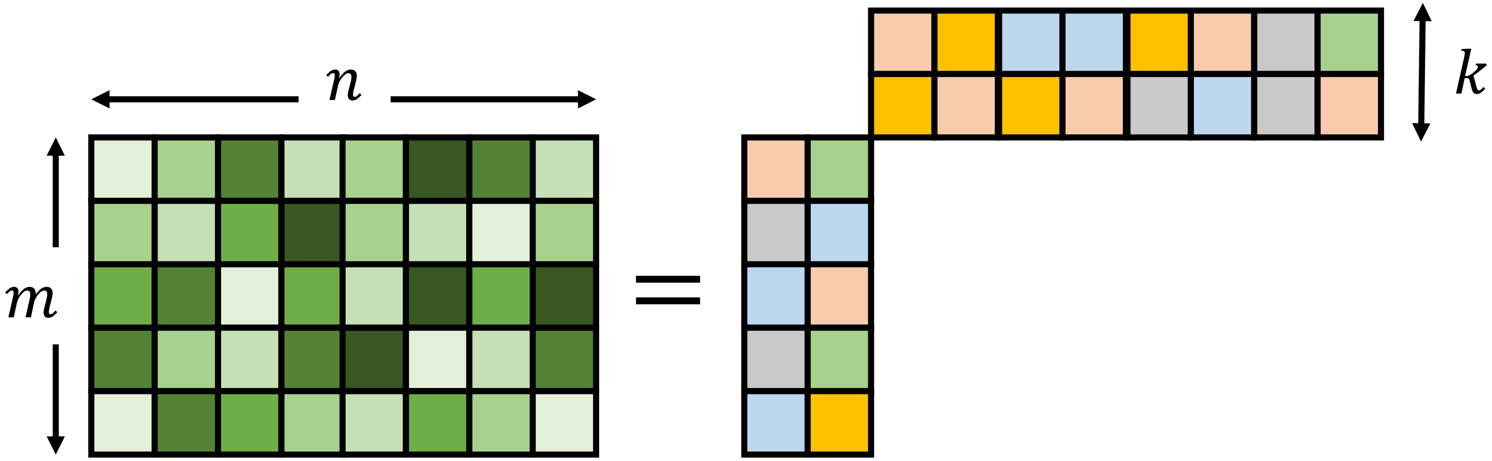


$p = 3, s = 2$

www.tes.com

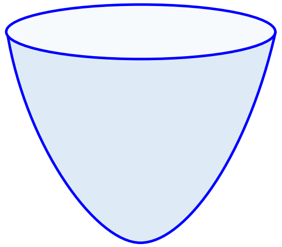
Recommender Systems





$$\min_{L\in\mathcal{M}_k^{m,n}} \|X_\Omega - L_\Omega\|_F^2$$

f



$\mathcal{C}$

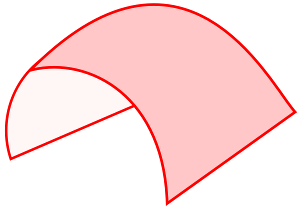
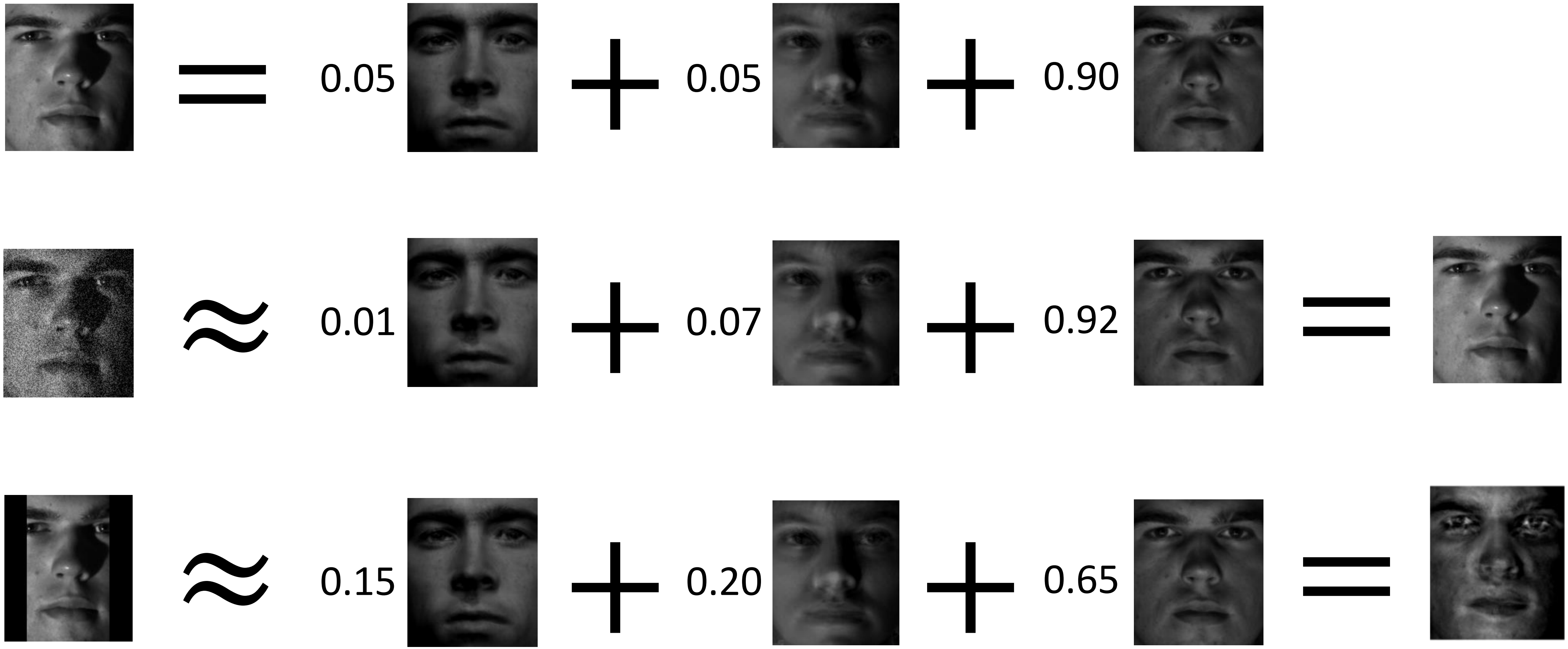


Image Reconstruction and Robust Face Recognition

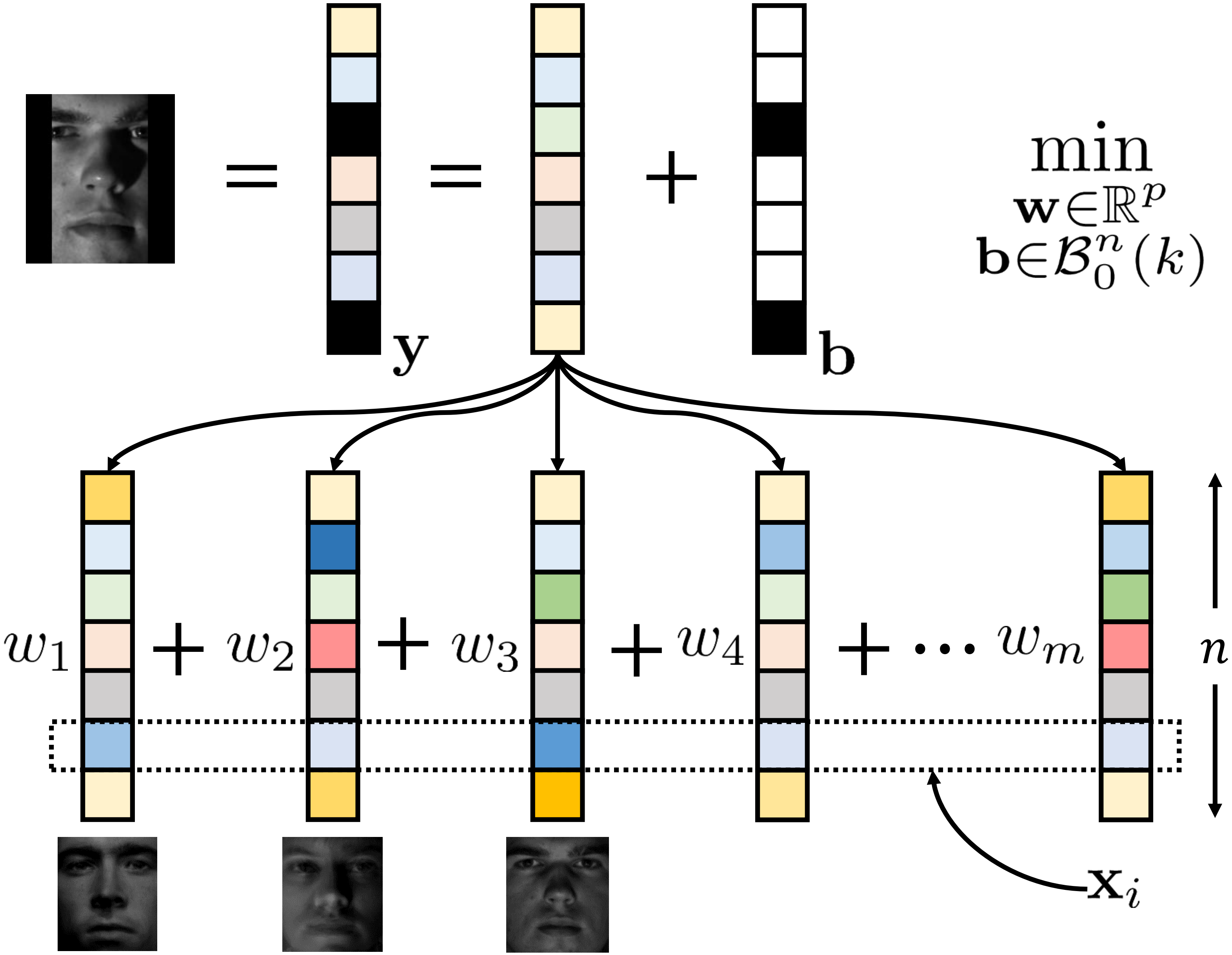


= 0.05 + 0.05 + 0.90

≈ 0.01 + 0.07 + 0.92 =

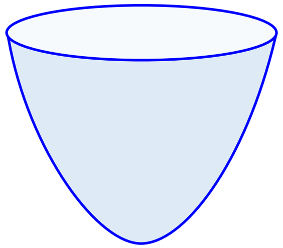
≈ 0.15 + 0.20 + 0.65 =

Image Denoising and Robust Face Recognition

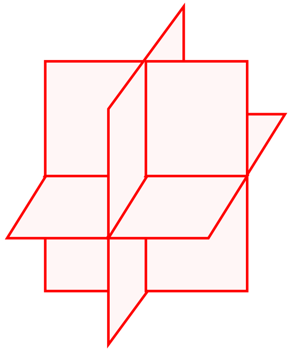


$$\min_{\substack{\mathbf{w}\in\mathbb{R}^p \\ \mathbf{b}\in\mathcal{B}_0^n(k)}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w} - b_i)^2$$

f



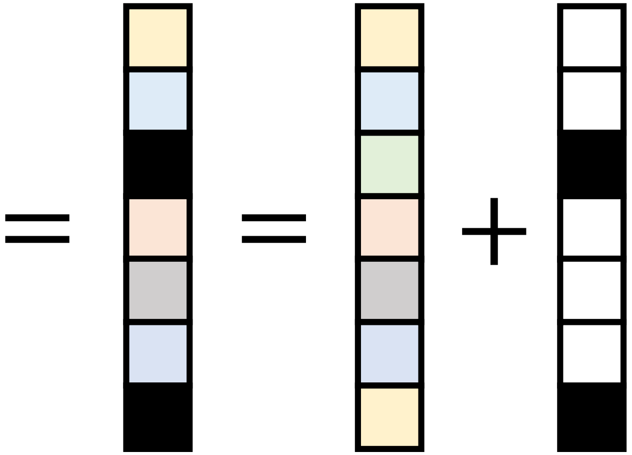
$\mathcal{C}$



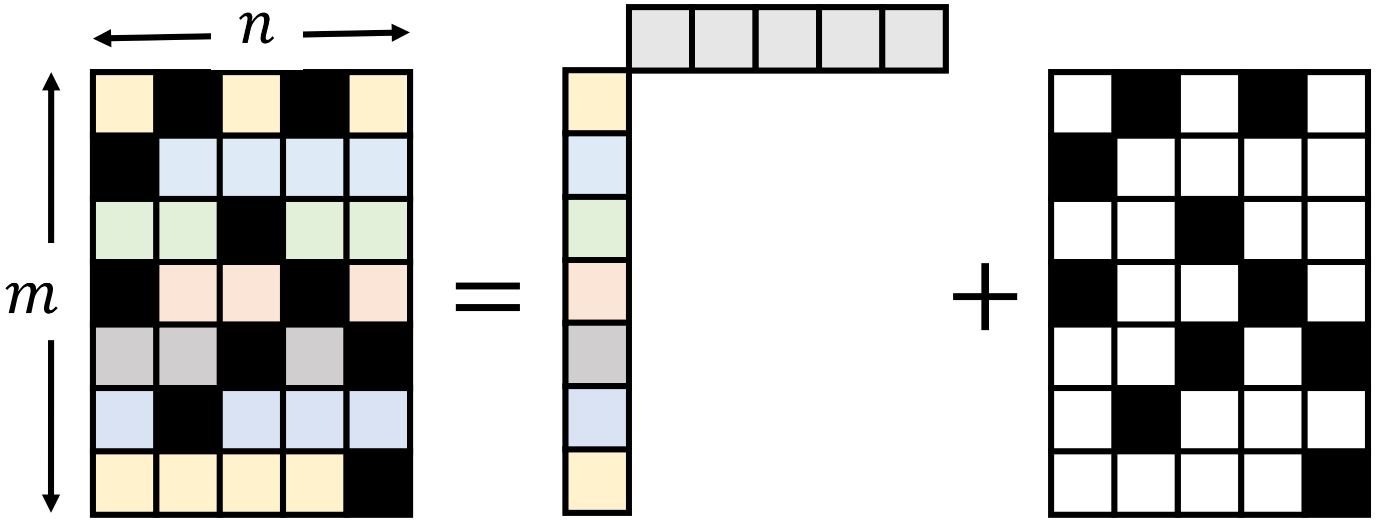
Large Scale Surveillance

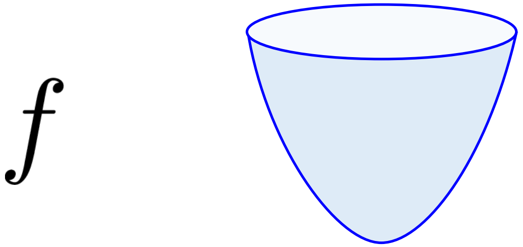
- Foreground-background separation

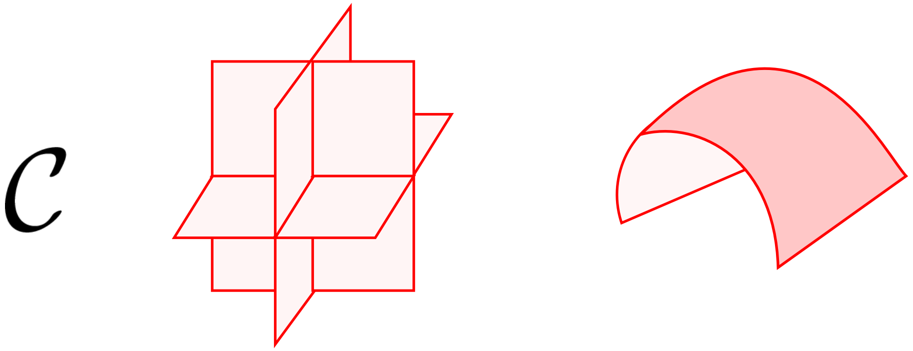
www.extremetech.com



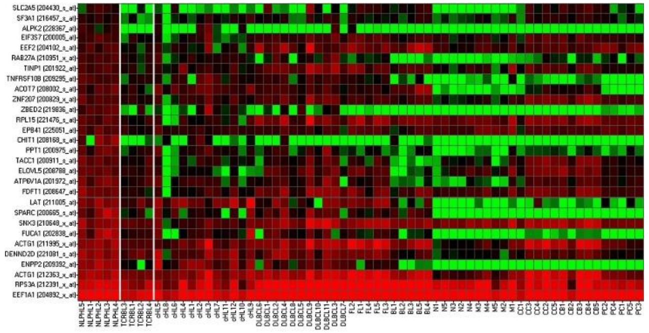
$$\min_{\substack{L\in\mathcal{M}_k^{m,n}\\ S\in\mathcal{B}_0^{m,n}(s)}} \|X-(L+S)\|_F^2$$







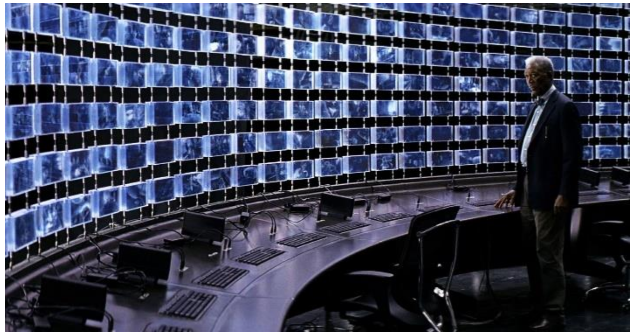
Non Convex Optimization



Sparse Recovery

Matrix Completion

Robust Regression



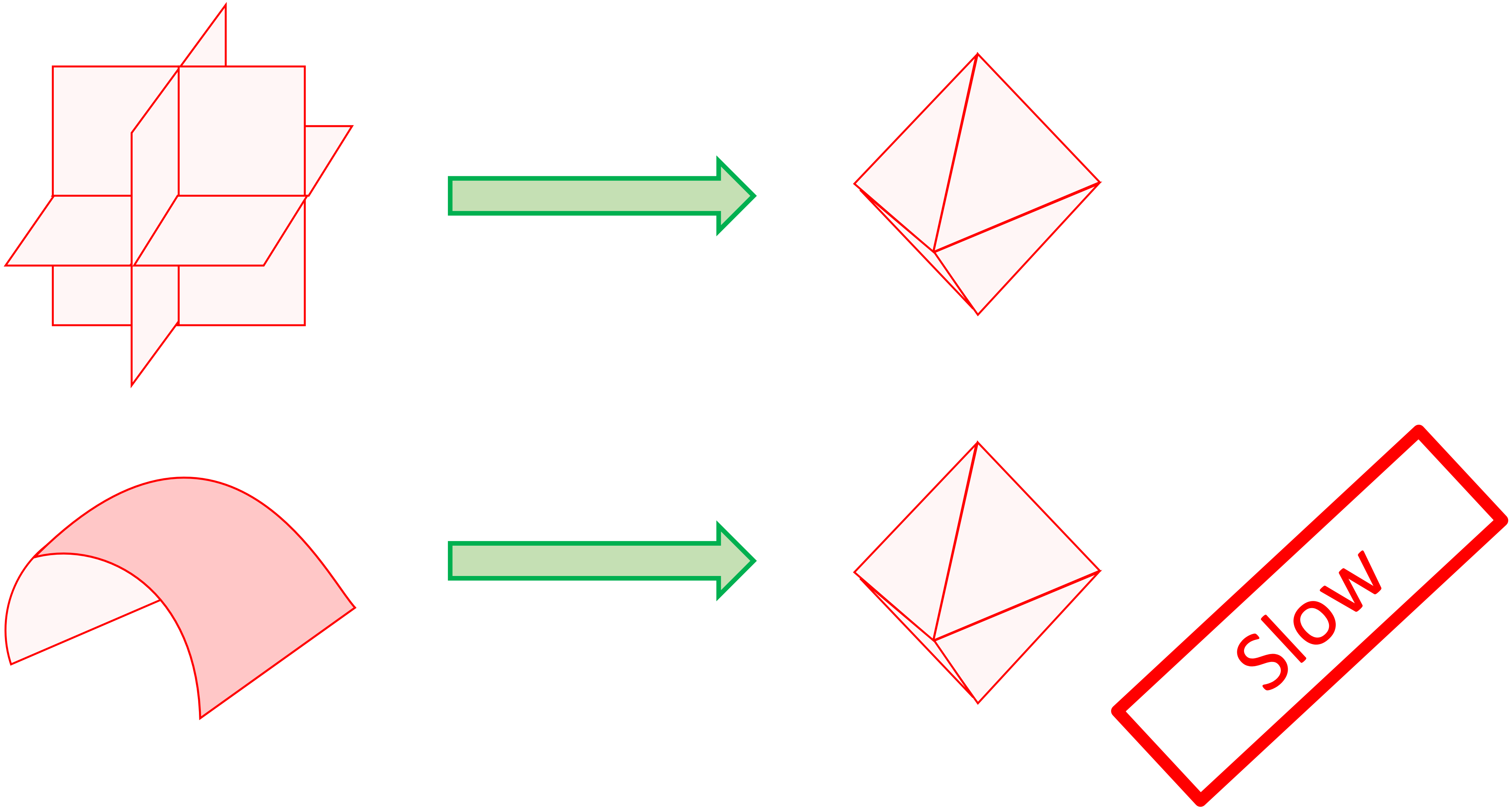
Robust PCA

NP-hard

Non-convex Optimization:
A Brief Introduction

Relaxation-based Techniques

- “Convexify” the feasible set



Alternating Minimization

$$\begin{aligned} \min \ & f(\mathbf{x}, \mathbf{y}) \\ s.t. \ & \mathbf{x} \in \mathcal{C}_1 \\ & \mathbf{y} \in \mathcal{C}_2 \end{aligned}$$

```
▷ Initialize $\mathbf{x}^0, \mathbf{y}^0$
▷ For $t = 1, 2, \ldots$
    ▷ $\mathbf{x}^t = \arg\min_{\mathbf{x} \in \mathcal{C}_1} f(\mathbf{x}, \mathbf{y}^{t-1})$
    ▷ $\mathbf{y}^t = \arg\min_{\mathbf{y} \in \mathcal{C}_2} f(\mathbf{x}^t, \mathbf{y})$
```

Matrix Completion

$$\min_{L \in \mathcal{M}_k^{m,n}} \|X_\Omega - L_\Omega\|_F^2$$

$$\equiv \min_{\substack{U \in \mathbb{R}^{m \times k} \\ V \in \mathbb{R}^{n \times k}}} \|X_\Omega - (UV^\top)_\Omega\|_F^2$$

Robust PCA

$$\min_{\substack{L \in \mathcal{M}_k^{m,n} \\ S \in \mathcal{B}_0^{m,n}(s)}} \|X - (L + S)\|_F^2$$

… also Robust Regression, coming up

Projected Gradient Descent

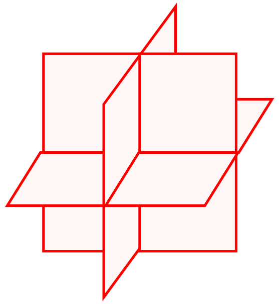
$$\min \; f(\mathbf{x})$$
$$s.t. \; \mathbf{x} \in \mathcal{C}$$

```
▷ Initialize $\mathbf{x}^0$
▷ For $t = 1, 2, \ldots$
    ▷ $\mathbf{z}^t = \mathbf{x}^{t-1} - \eta_t \cdot \nabla f(\mathbf{x}^{t-1}))$
    ▷ $\mathbf{x}^t = \Pi_{\mathcal{C}}(\mathbf{z}^t)$
```

$$\Pi_{\mathcal{C}}(\mathbf{z}) = \arg\min_{\mathbf{x} \in \mathcal{C}} \|\mathbf{z} - \mathbf{x}\|_2^2$$

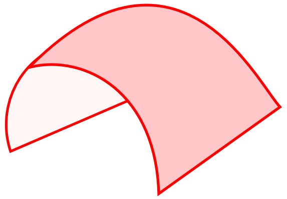
Non-convex Projection

$\mathcal{B}_0^p(s)$



Top s elements by magnitude

$\mathcal{M}_k^{m,n}$



Perform k-truncated SVD

Sparse Recovery

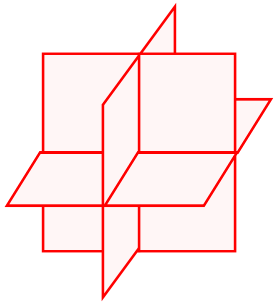
$$\min_{\mathbf{w} \in \mathcal{B}_0^p(s)} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w})^2$$

Pursuit and Greedy Methods

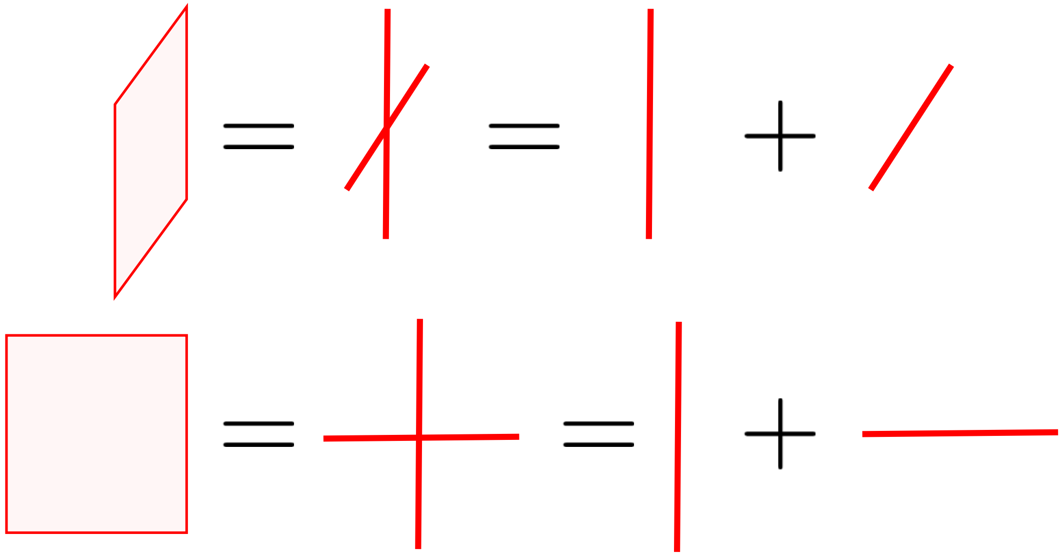
$$\min \; f(\mathbf{x})$$
$$s.t. \; \mathbf{x} \in \mathcal{C}$$

$\mathcal{A}$ Set of “atoms”

$$\mathcal{C} = \left\{ \mathbf{x} = \sum_{i=1}^{s} \mathbf{a}_i : \mathbf{a}_i \in \mathcal{A} \right\}$$



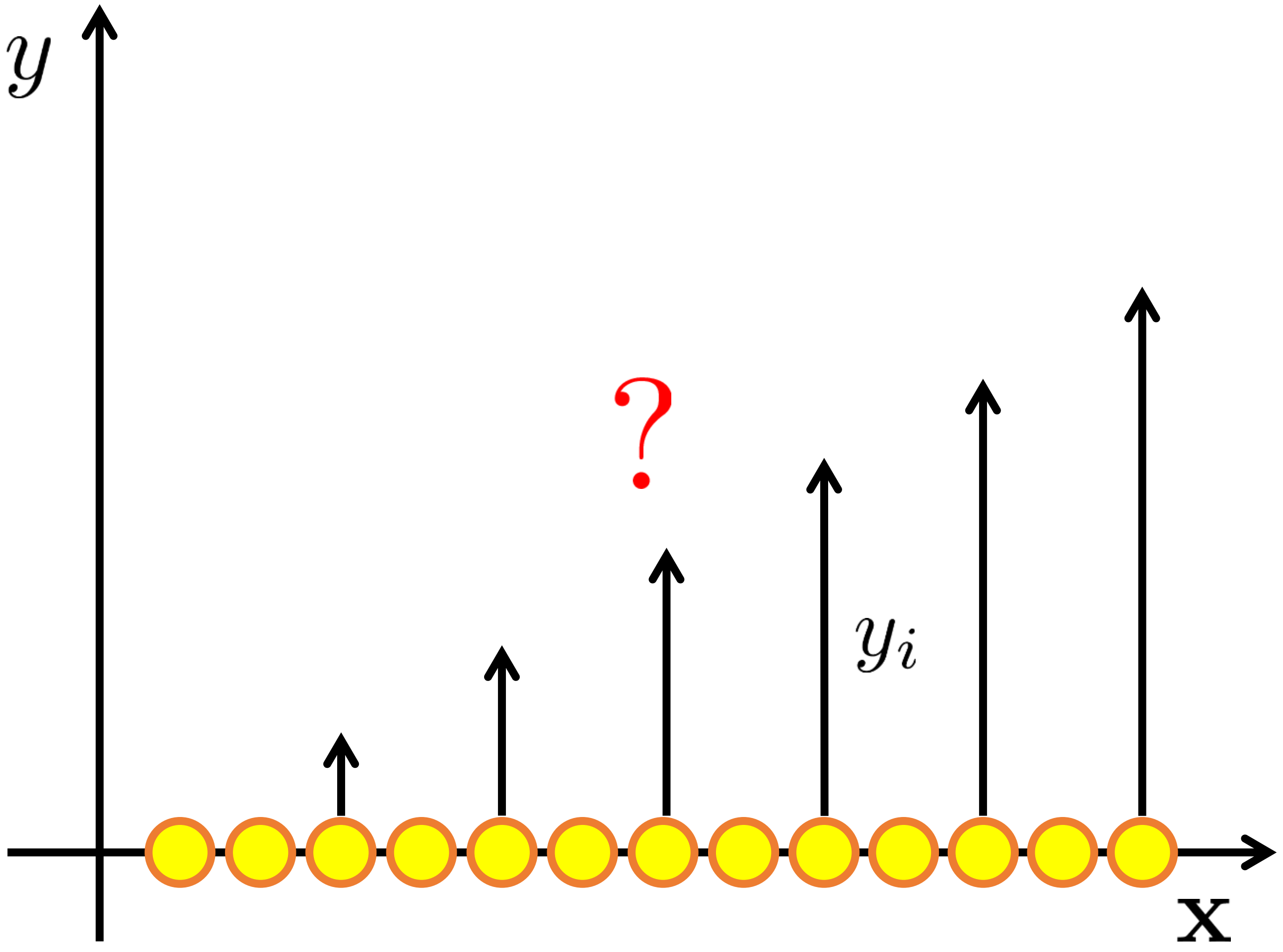
Sparse Recovery



```
▷ Initialize $S^0 = \phi$
▷ For $t = 1, 2, \dots$
    ▷ $\mathbf{a}^t =$ “best” greedy choice
    ▷ $S^t = S^{t-1} \cup \{\mathbf{a}^t\}$
    ▷ $\mathbf{x}^t = \arg\min_{\mathbf{x} \in \mathrm{conv}(S^t)} f(\mathbf{x})$
```

Robust Regression:
A Non-convex Approach

Linear Regression



Data: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^p$

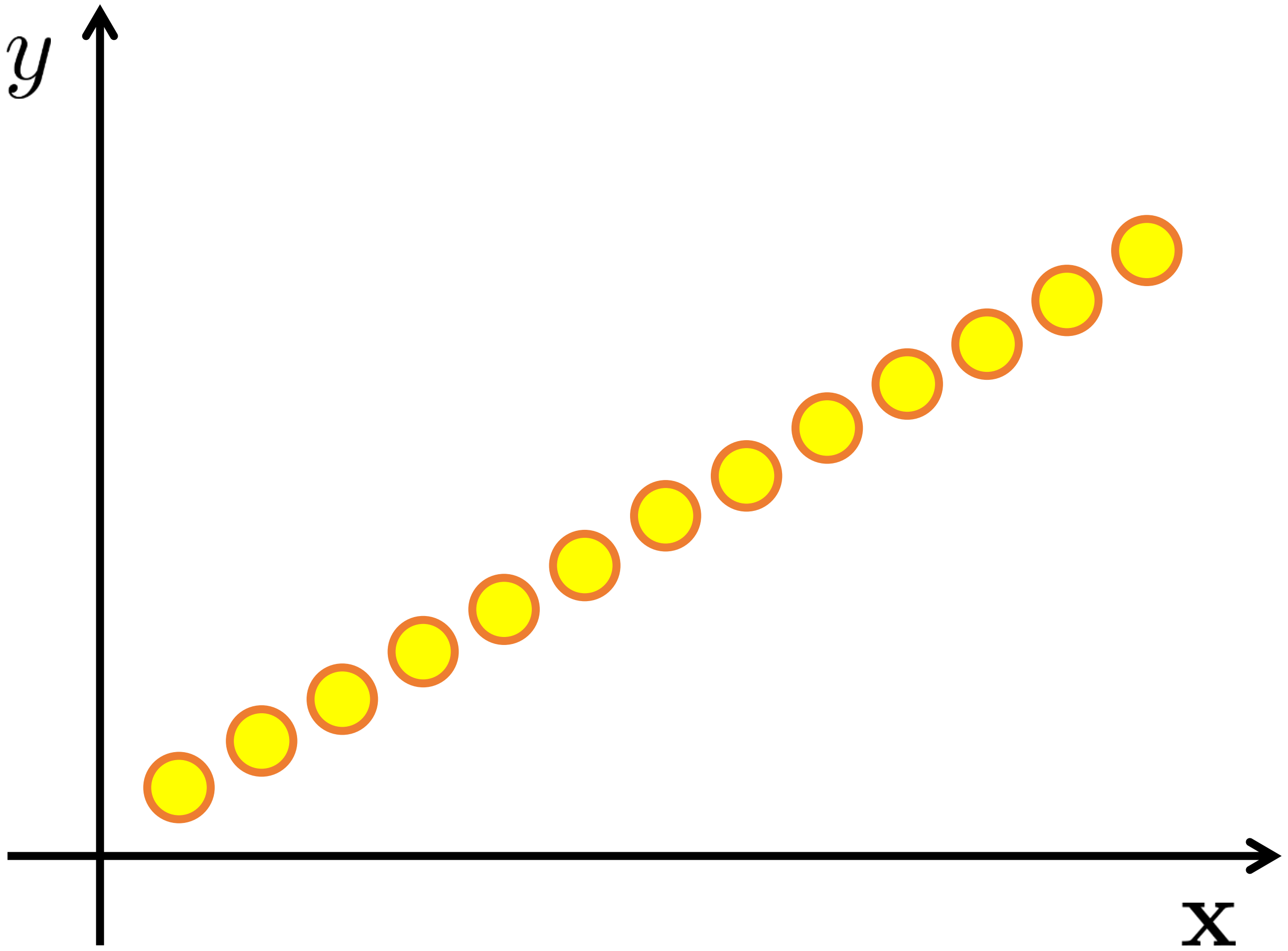
Model: $\mathbf{w}^*$ (hidden)

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle$$

Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

Recover $\mathbf{w}^*$?

Linear Regression

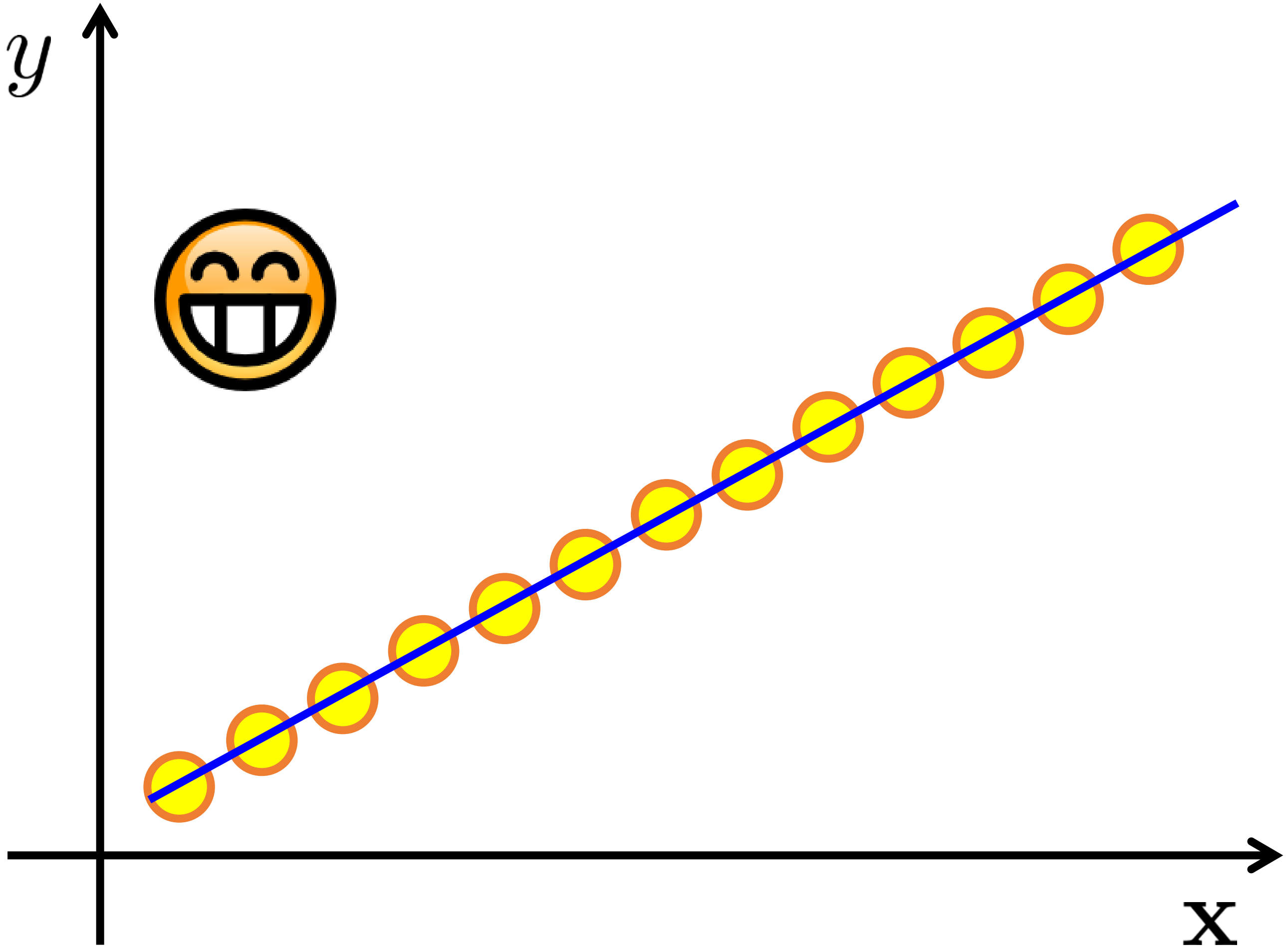


$$\text{Given: } (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle$$

$$n \left\updownarrow \vphantom{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}} \right. \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} = \underbrace{\overset{\overset{p}{\longleftrightarrow}}{\begin{bmatrix} — \; \mathbf{x}_1 \; — \\ — \; \mathbf{x}_2 \; — \\ — \; \mathbf{x}_3 \; — \\ \vdots \\ — \; \mathbf{x}_n \; — \end{bmatrix}}}_{\mathbf{X}} \underbrace{\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix}}_{?} \left. \vphantom{\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix}} \right\updownarrow p$$

Linear Regression



Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

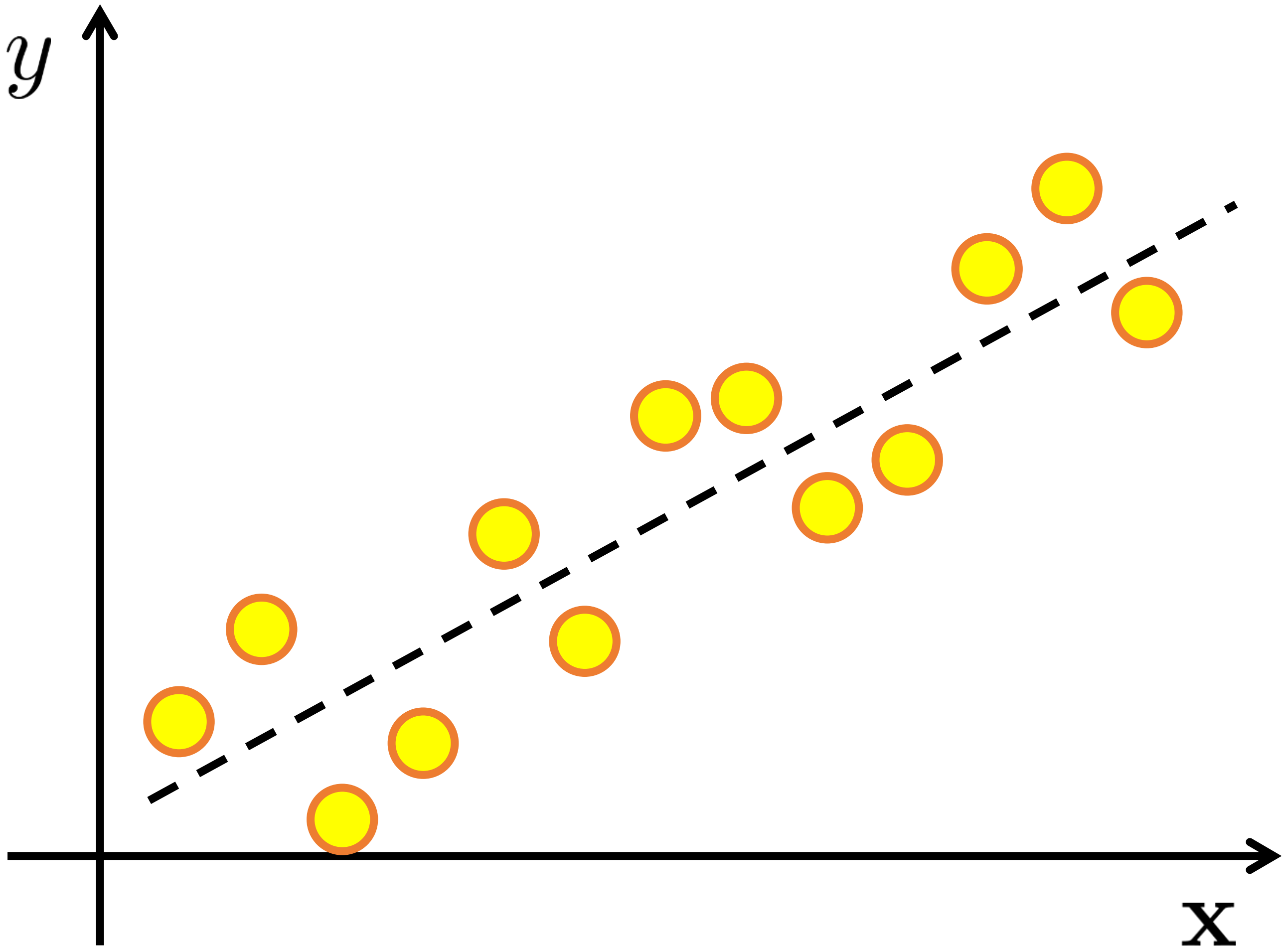
$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle$$

$$\mathbf{y} = \mathbf{X}\mathbf{w}$$

Linear system!!

$\mathbf{w}^*$ Recovered!!

Linear Regression with Noise

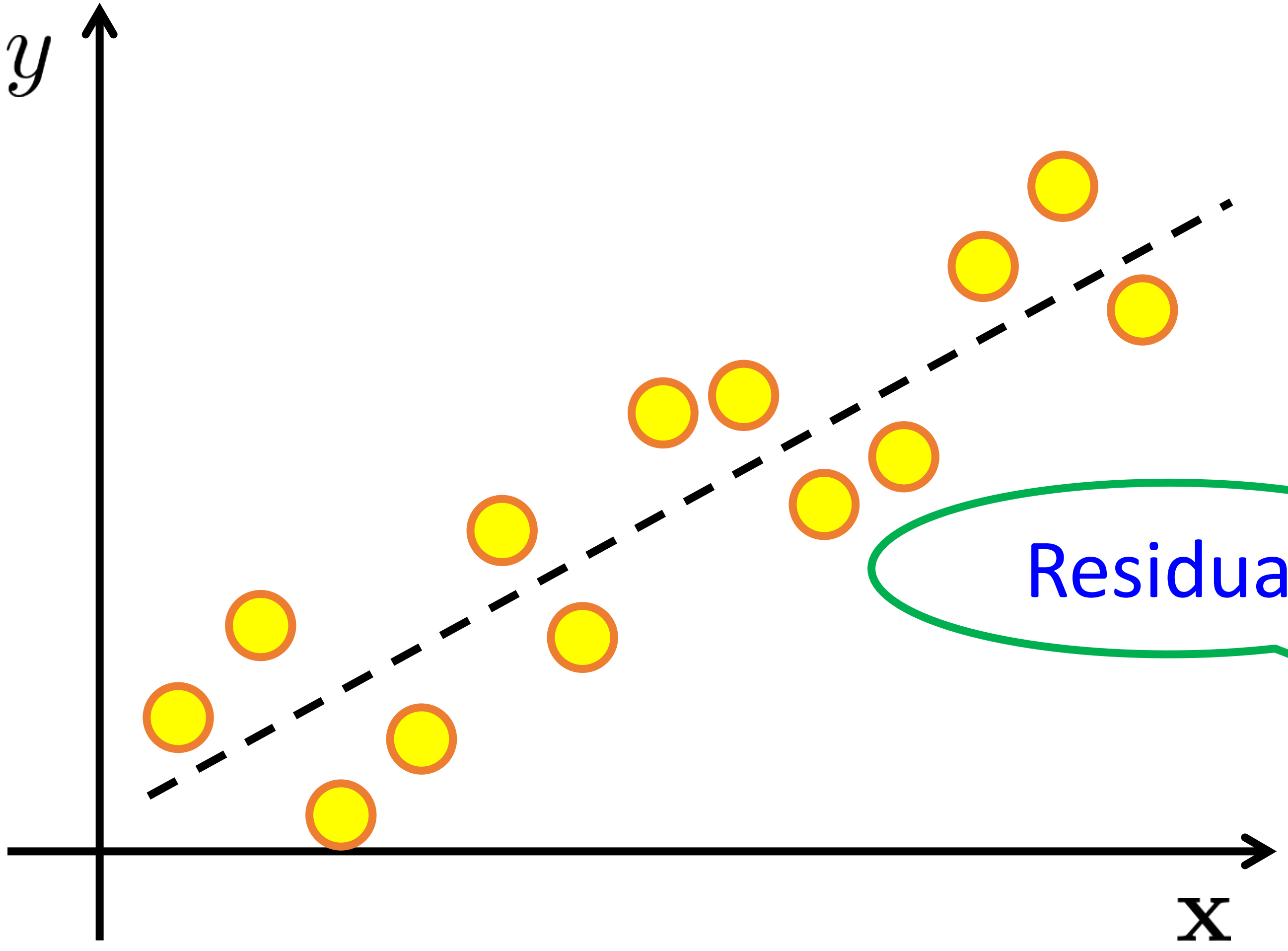


Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i$$

$$n \updownarrow \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} \approx \underbrace{\overset{\overset{p}{\longleftrightarrow}}{\begin{bmatrix} \text{——} \ \mathbf{x}_1 \ \text{——} \\ \text{——} \ \mathbf{x}_2 \ \text{——} \\ \text{——} \ \mathbf{x}_3 \ \text{——} \\ \vdots \\ \text{——} \ \mathbf{x}_n \ \text{——} \end{bmatrix}}}_{\mathbf{X}} \underset{?}{\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix}} \updownarrow p$$

Linear Regression with Noise



Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i$$

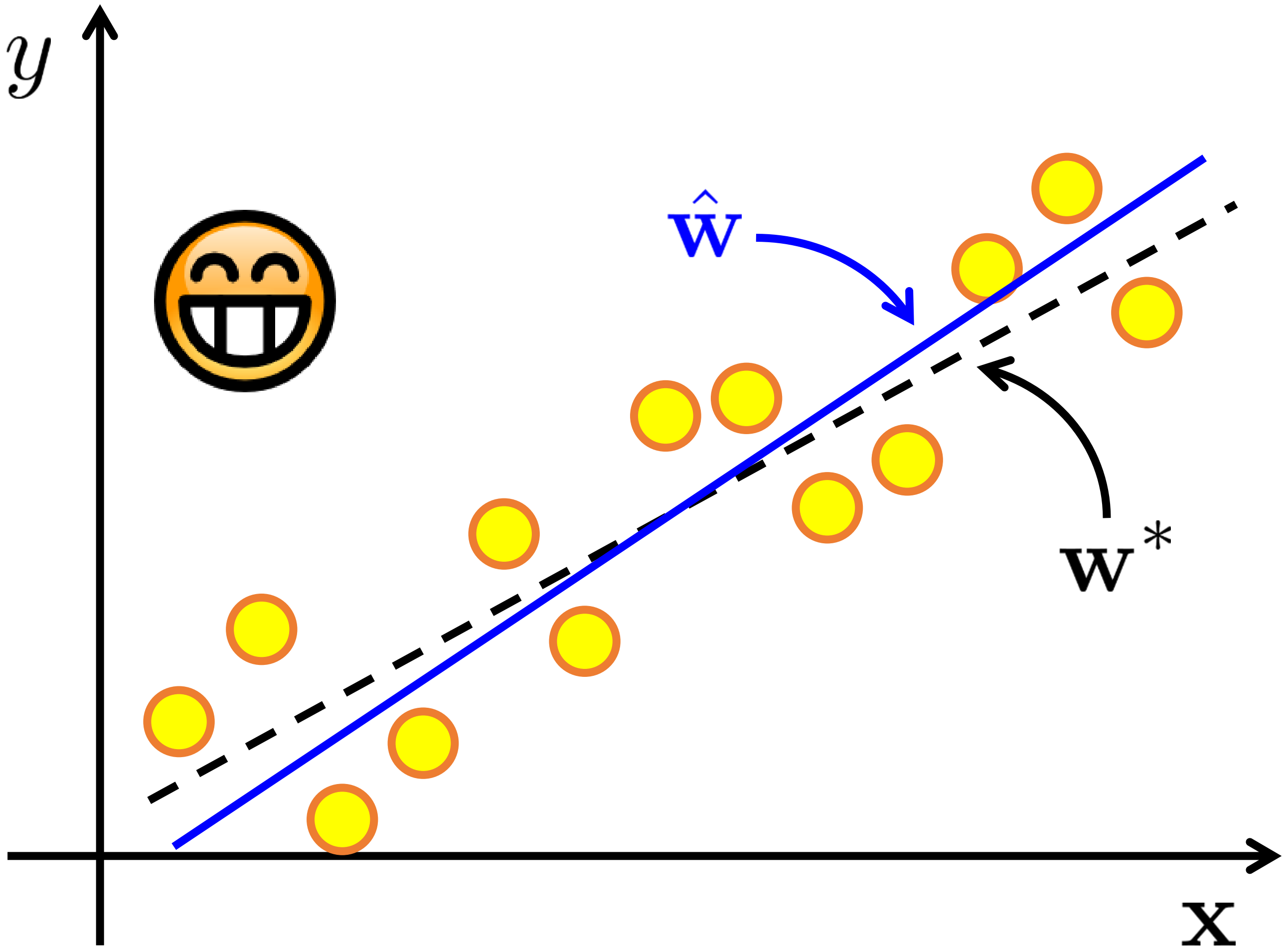
$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w}^*$$

$\mathbf{e}$ is "small"

Residual

find $\mathbf{w}$ that guarantees small $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$

Linear Regression with Noise



Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

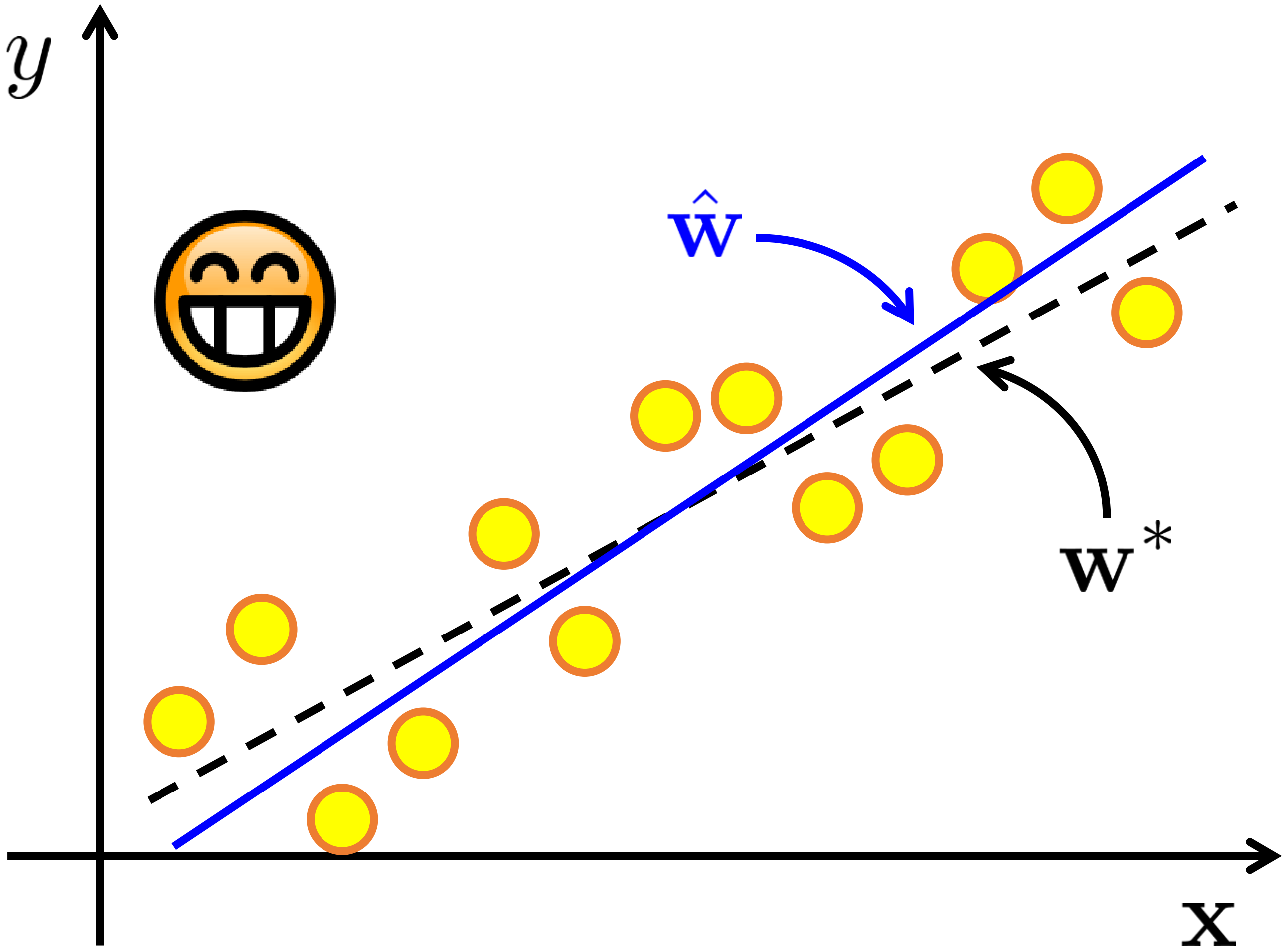
$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i$$

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2 + \lambda \|\mathbf{w}\|_2^2$$

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X} + \lambda I\right)^{-1} \mathbf{X}^\top \mathbf{y}$$

$\mathbf{w}^*$ Recovered!!

Linear Regression with Noise



Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

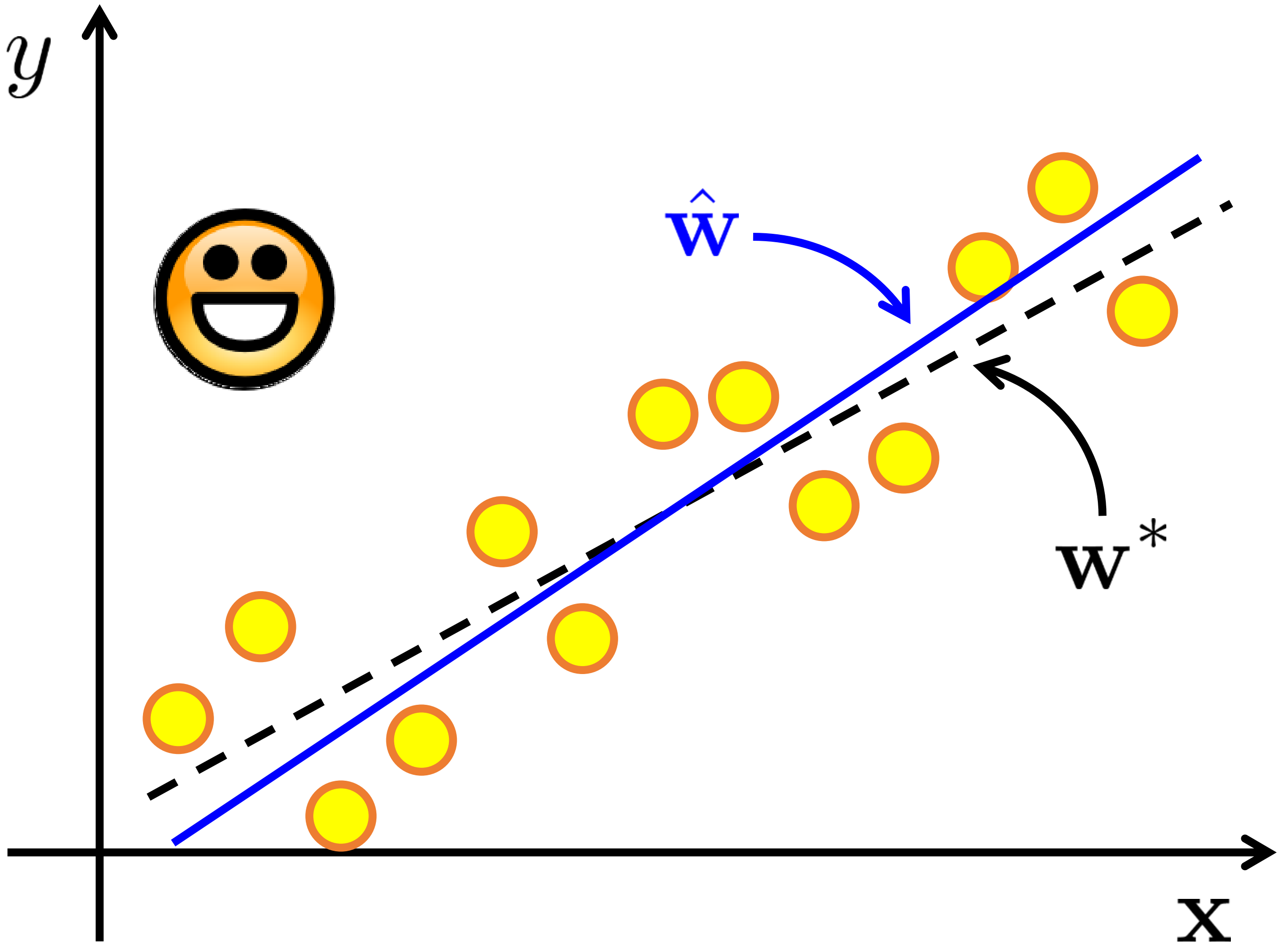
$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i$$

If $e_i \sim \mathcal{N}(0, \sigma^2)$

$$\|\hat{\mathbf{w}} - \mathbf{w}^*\|_2^2 \lesssim \frac{\sigma^2 p}{n}$$

$\mathbf{w}^*$ Recovered!!

Linear Regression with Noise



Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i$$

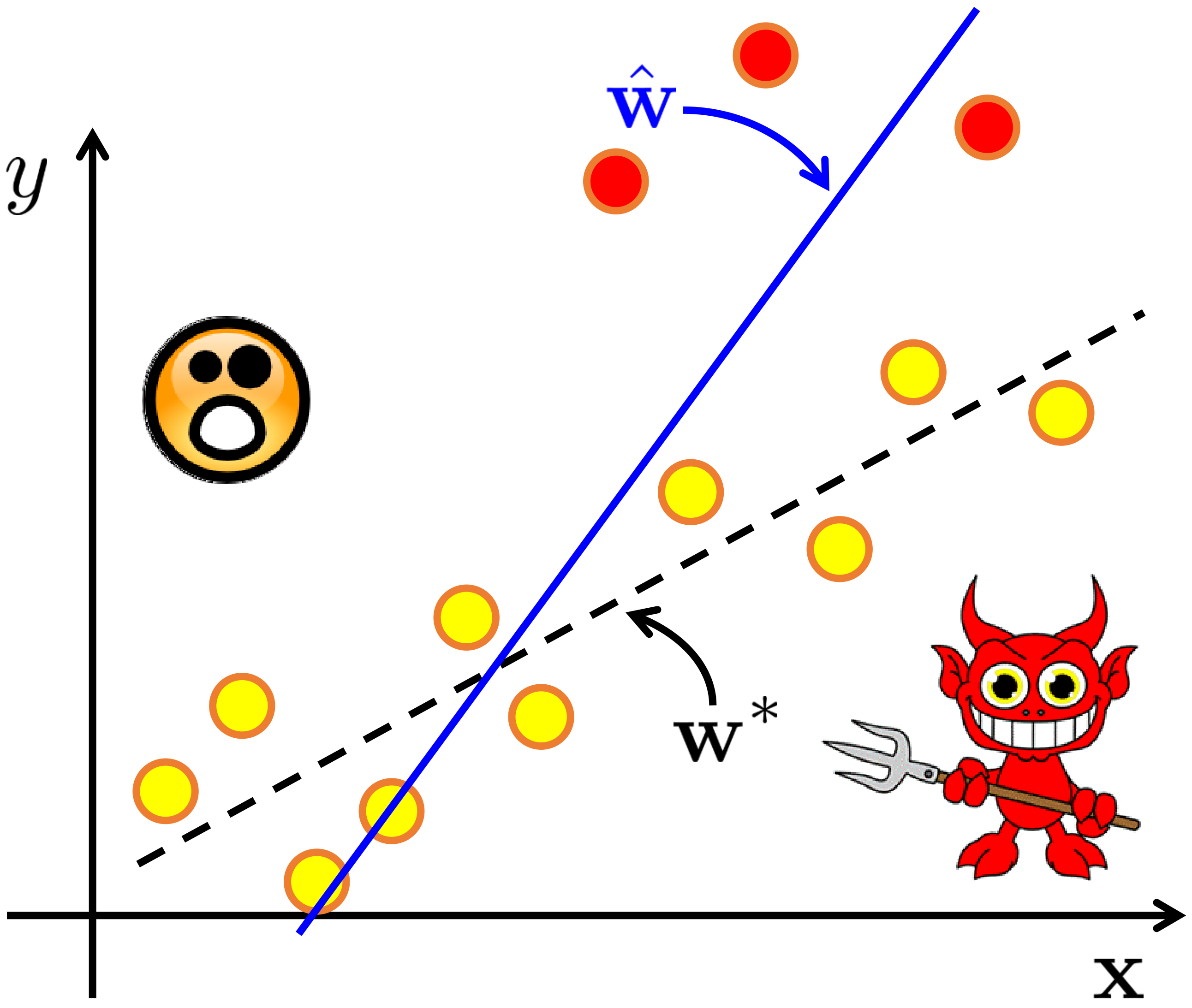
If $e_i \sim \mathcal{N}(0, \sigma^2)$

$$\|\hat{\mathbf{w}} - \mathbf{w}^*\|_2^2 \lesssim \frac{\sigma^2 p}{n}$$

$\mathbf{w}^*$ Recovered!!

(almost)

Linear Regression with Corruptions



www.toonvectors.com

Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i + b_i$$

No $\mathbf{w}$ can guarantee small $\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2$

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X} + \lambda I\right)^{-1} \mathbf{X}^\top \mathbf{y}$$

Still recover $\mathbf{w}^*$?

Robust Regression

$$\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{b}$$

Corruptions are adversarial, adaptive, but only on a “few” locations

$$\|\mathbf{b}\|_0 \leq k = \alpha \cdot n$$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} \text{---} & \mathbf{x}_1 & \text{---} \\ \text{---} & \mathbf{x}_2 & \text{---} \\ \text{---} & \mathbf{x}_3 & \text{---} \\ \text{---} & \mathbf{x}_4 & \text{---} \\ \text{---} & \mathbf{x}_5 & \text{---} \\ \text{---} & \mathbf{x}_6 & \text{---} \end{bmatrix}}_{\mathbf{X}} \begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ b_2 \\ 0 \\ 0 \\ b_5 \\ 0 \end{bmatrix}}_{\mathbf{b}}$$

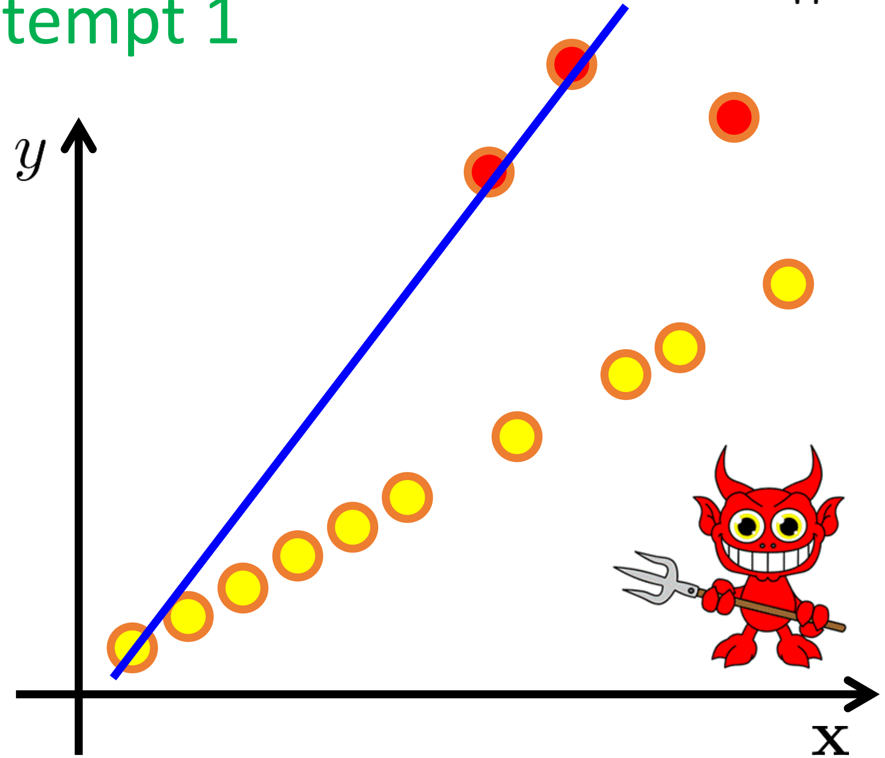
Robust Regression

$$\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{b}$$

Corruptions are adversarial, adaptive, but only on a “few” locations

$$\|\mathbf{b}\|_0 \leq k = \alpha \cdot n$$

Attempt 1



3

$$\mathbf{b} = \mathbf{y} - \mathbf{X}\mathbf{w}$$

$$\min \|\mathbf{y} - \mathbf{X}\mathbf{w} - \mathbf{b}\|_2^2$$

$$s.t.\ \|\mathbf{b}\|_0 \leq k$$

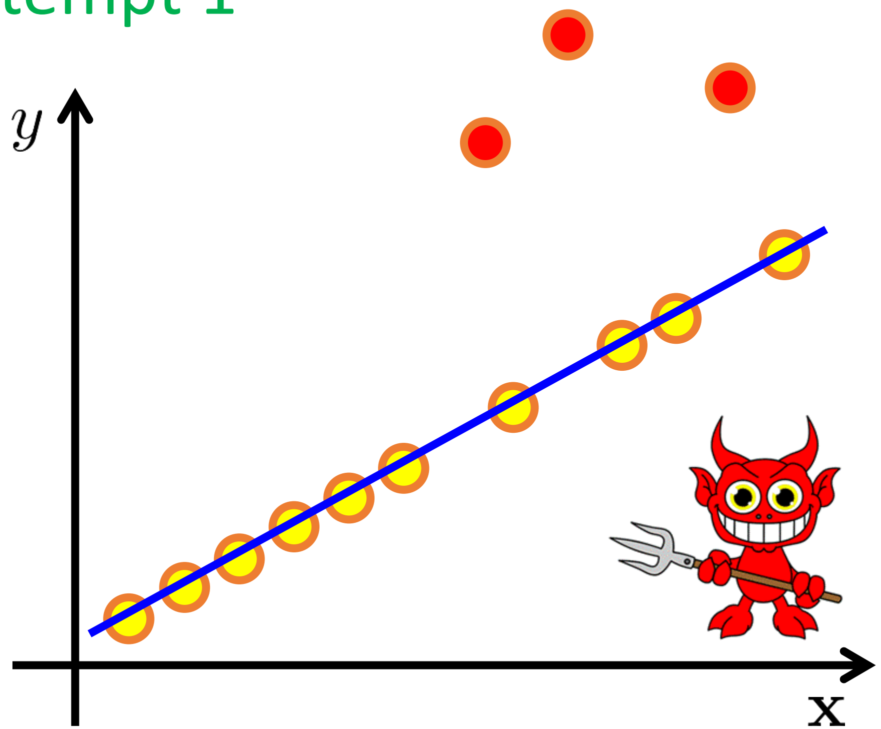
Robust Regression

$$\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{b}$$

Corruptions are adversarial, adaptive, but only on a “few” locations

$$\|\mathbf{b}\|_0 \leq k = \alpha \cdot n$$

Attempt 1



10

$$\mathbf{b} = \mathbf{y} - \mathbf{X}\mathbf{w}$$

$$\min \|\mathbf{y} - \mathbf{X}\mathbf{w} - \mathbf{b}\|_2^2$$

$$s.t. \ \|\mathbf{b}\|_0 \leq k$$

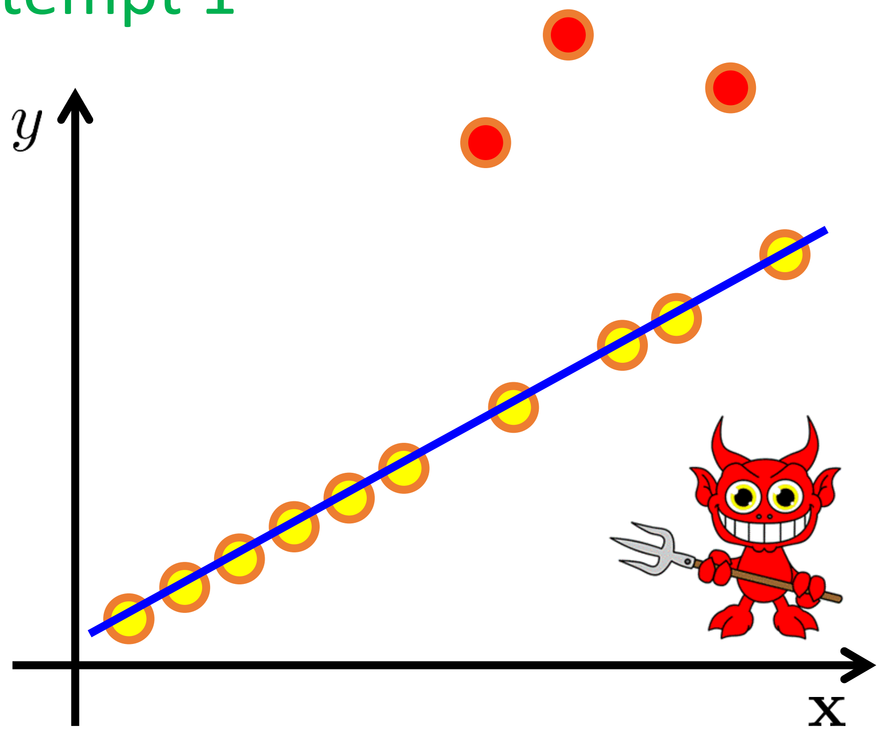
Robust Regression

$$\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{b}$$

Corruptions are adversarial, adaptive, but only on a “few” locations

$$\|\mathbf{b}\|_0 \leq k = \alpha \cdot n$$

Attempt 1



10

$$\mathbf{b} = \mathbf{y} - \mathbf{X}\mathbf{w}$$

$$\min \|\mathbf{y} - \mathbf{X}\mathbf{w} - \mathbf{b}\|_2^2$$
$$s.t. \; \|\mathbf{b}\|_0 \leq k$$

NP-hard!!

Robust Regression

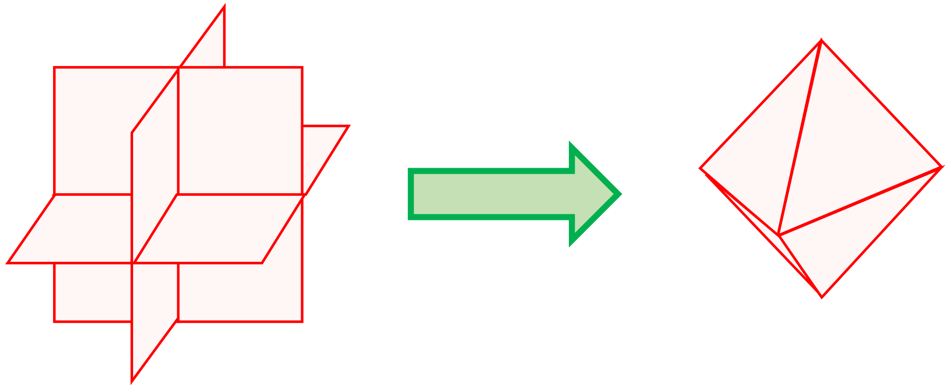
$$\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{b}$$

Corruptions are adversarial, adaptive, but only on a “few” locations

$$\|\mathbf{b}\|_0 \leq k = \alpha \cdot n$$

Attempt 2

$$\mathbf{b} = \mathbf{y} - \mathbf{X}\mathbf{w}$$



$$\min \ \|\mathbf{y} - \mathbf{X}\mathbf{w} - \mathbf{b}\|_2^2$$
$$s.t. \ \|\mathbf{b}\|_1 \leq \lambda$$

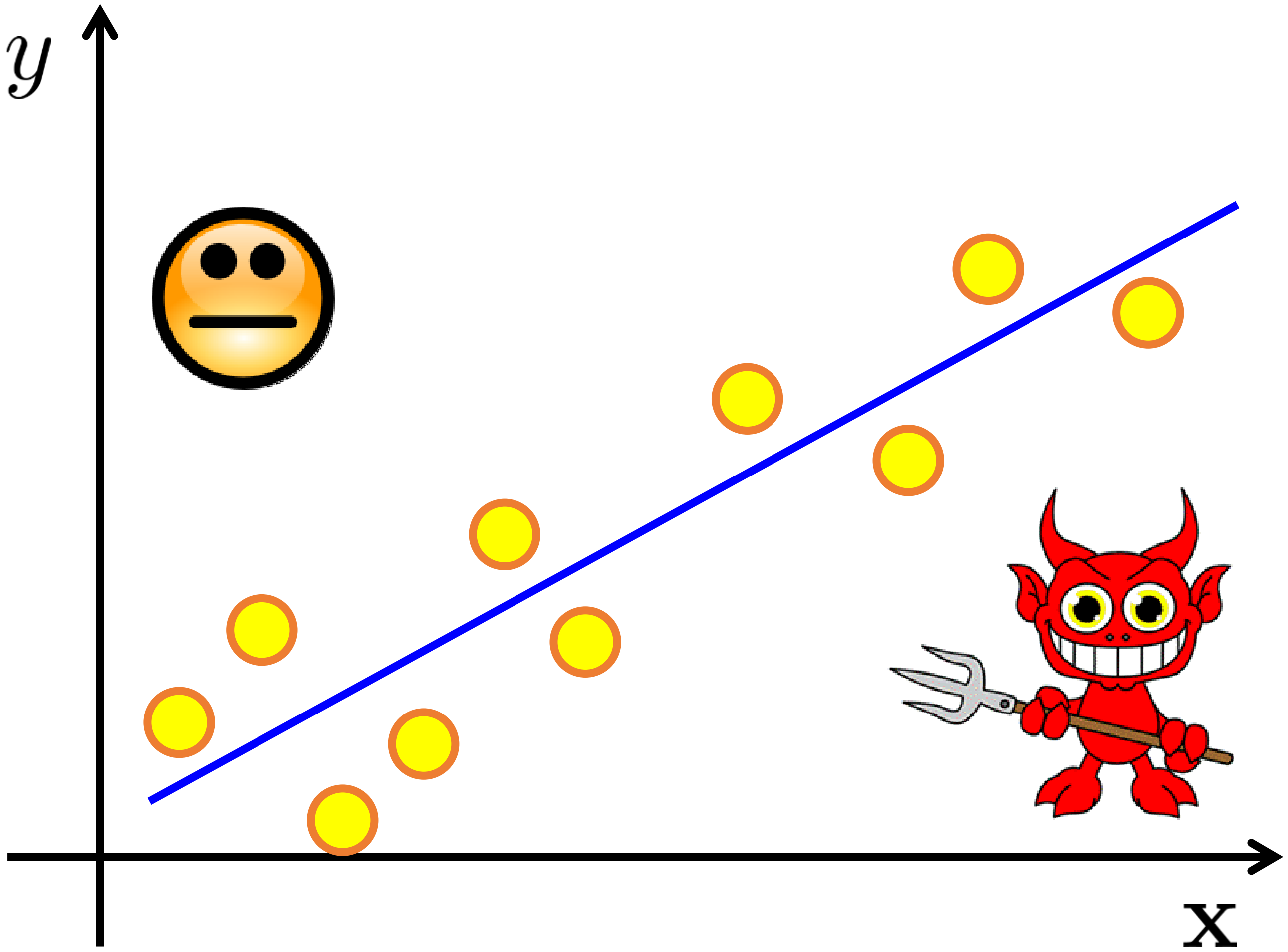
Expensive!!

[Wright and Ma 2010*, Nguyen et al., 2013*]

Lessons from History

> *If among these errors are some which appear too large to be admissible, then those equations which produced these errors will be rejected, as coming from too faulty experiments, and the unknowns will be determined by means of the other equations, which will then give much smaller errors*
>
> Adrien-Marie Legendre, *On the Method of Least Squares*, 1805

Linear Regression with Corruptions

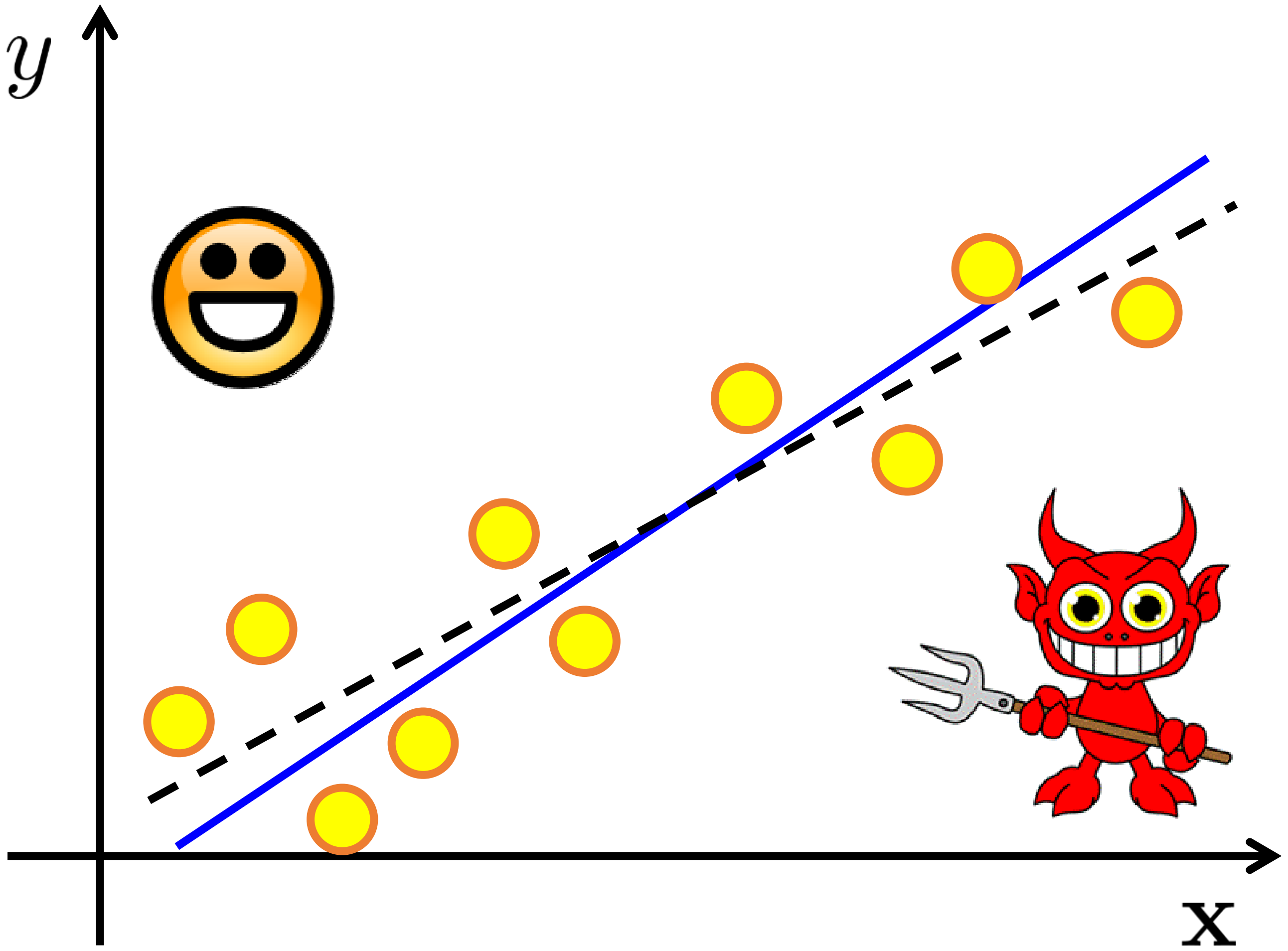


Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i + b_i$$

Given $\mathbf{w}^*$,
easy to identify ●

Linear Regression with Corruptions



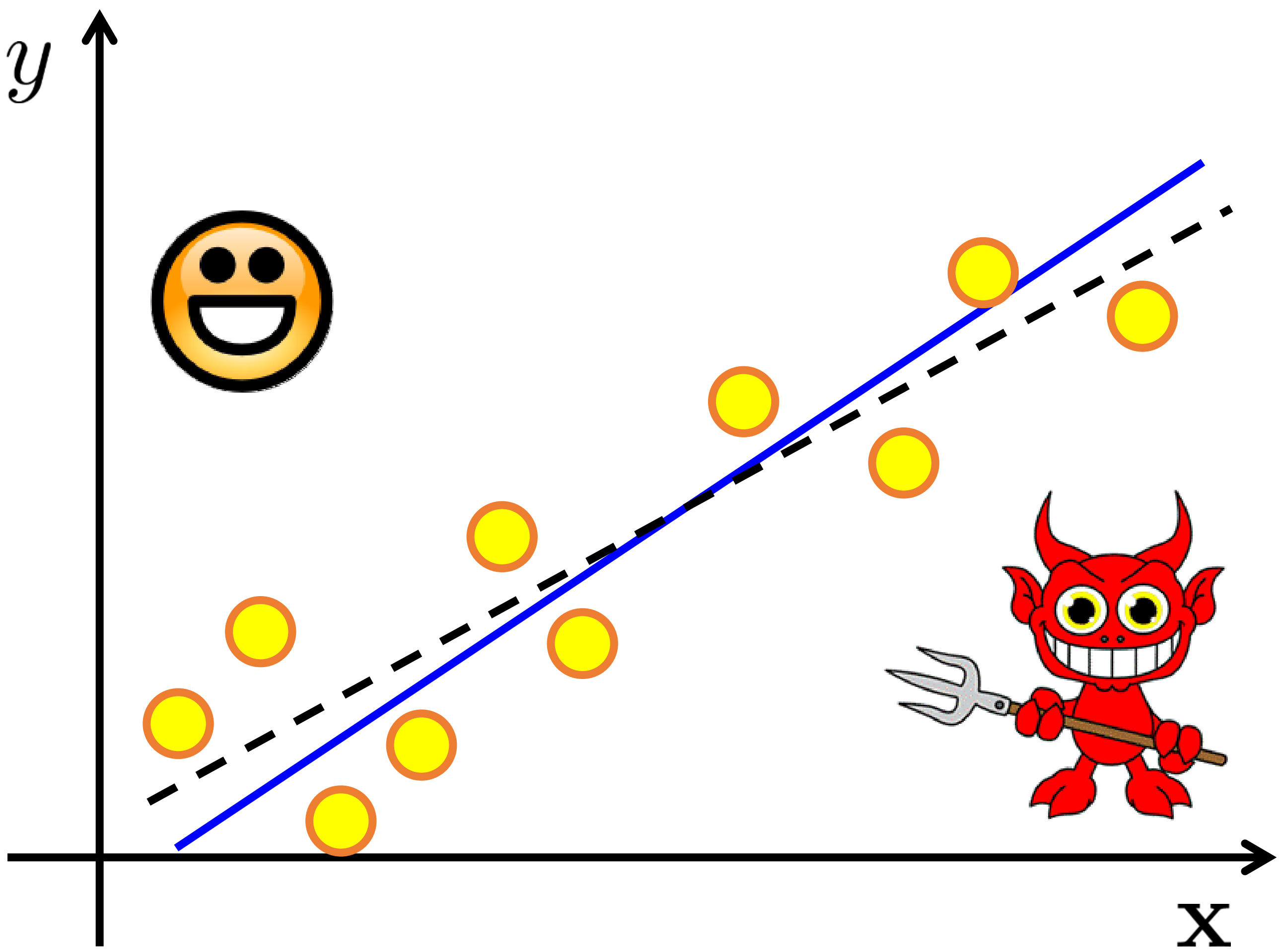
Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i + b_i$$

Given $\mathbf{w}^*$,
easy to identify ●

Given clean points,
$\mathbf{w}^*$ is recoverable

Linear Regression with Corruptions



Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i + b_i$$

Given $\mathbf{w}^*$,
easy to identify ●

Given clean points,
$\mathbf{w}^*$ is recoverable

Linear Regression with Corruptions

TORRENT-FC

Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i + b_i$$

Calculate $r_i = |y_i - \langle \hat{\mathbf{w}}, \mathbf{x}_i \rangle|$

Set aside k points with highest r_i

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

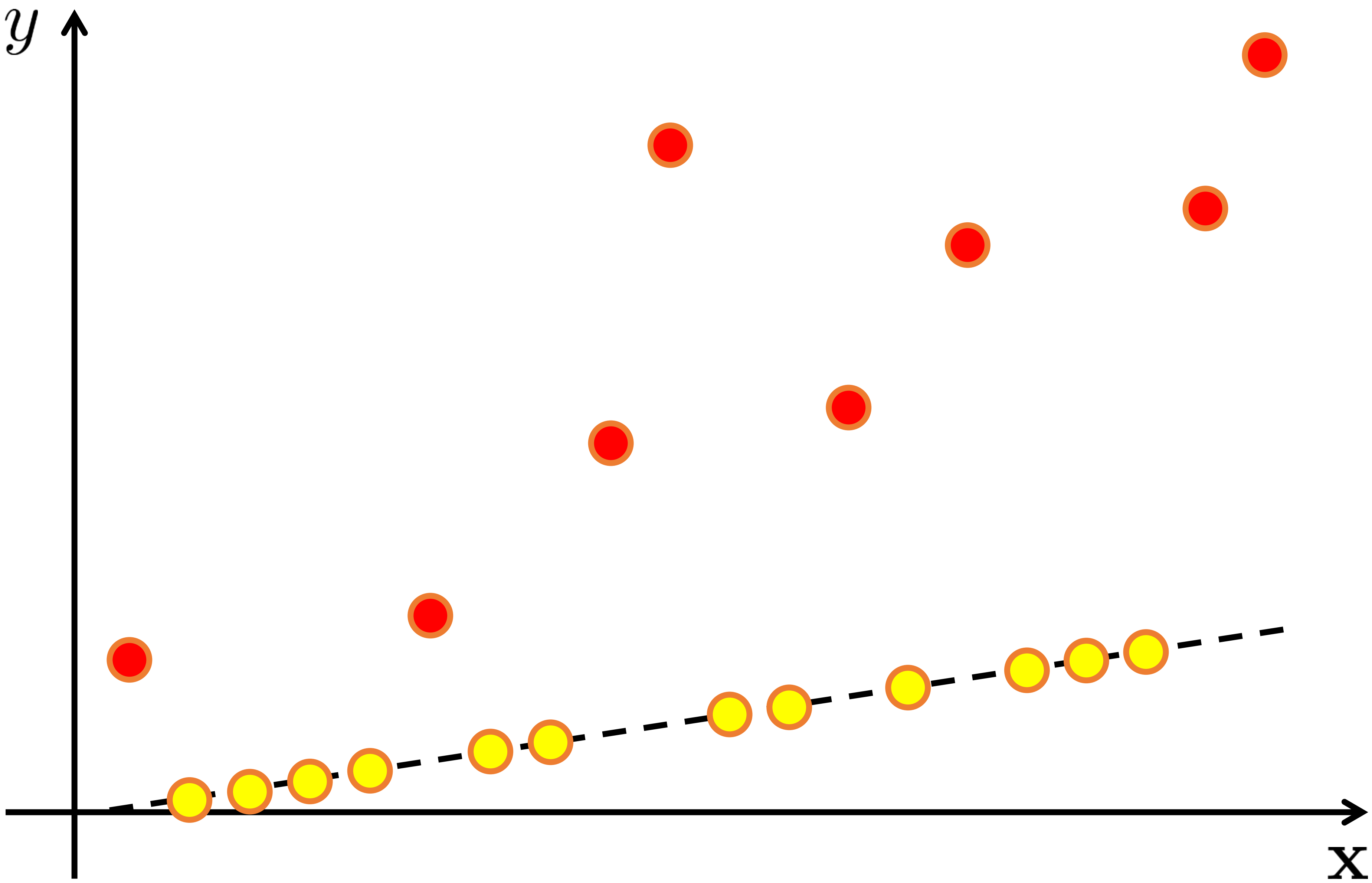
$\mathcal{A}$: active points

$$\hat{\mathbf{w}} = \min_{\mathbf{w}} \sum_{i \in \mathcal{A}} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2$$

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

Thresholding Operator-based Robust RegrEssioN meThod [Bhatia *et al*, 2015]

TORRENT in Action!

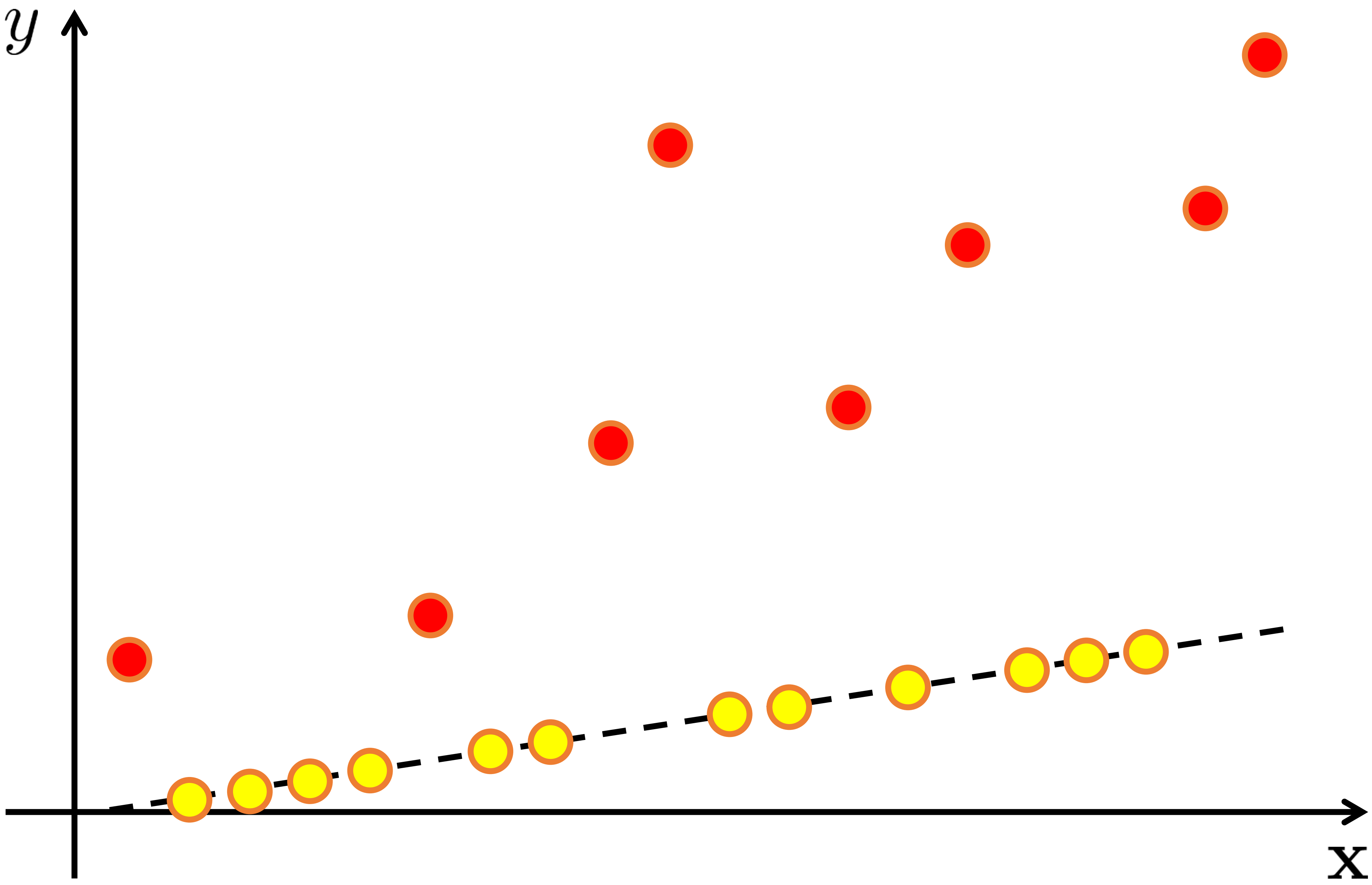


$$\mathbf{w}^* = 2.5$$

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!

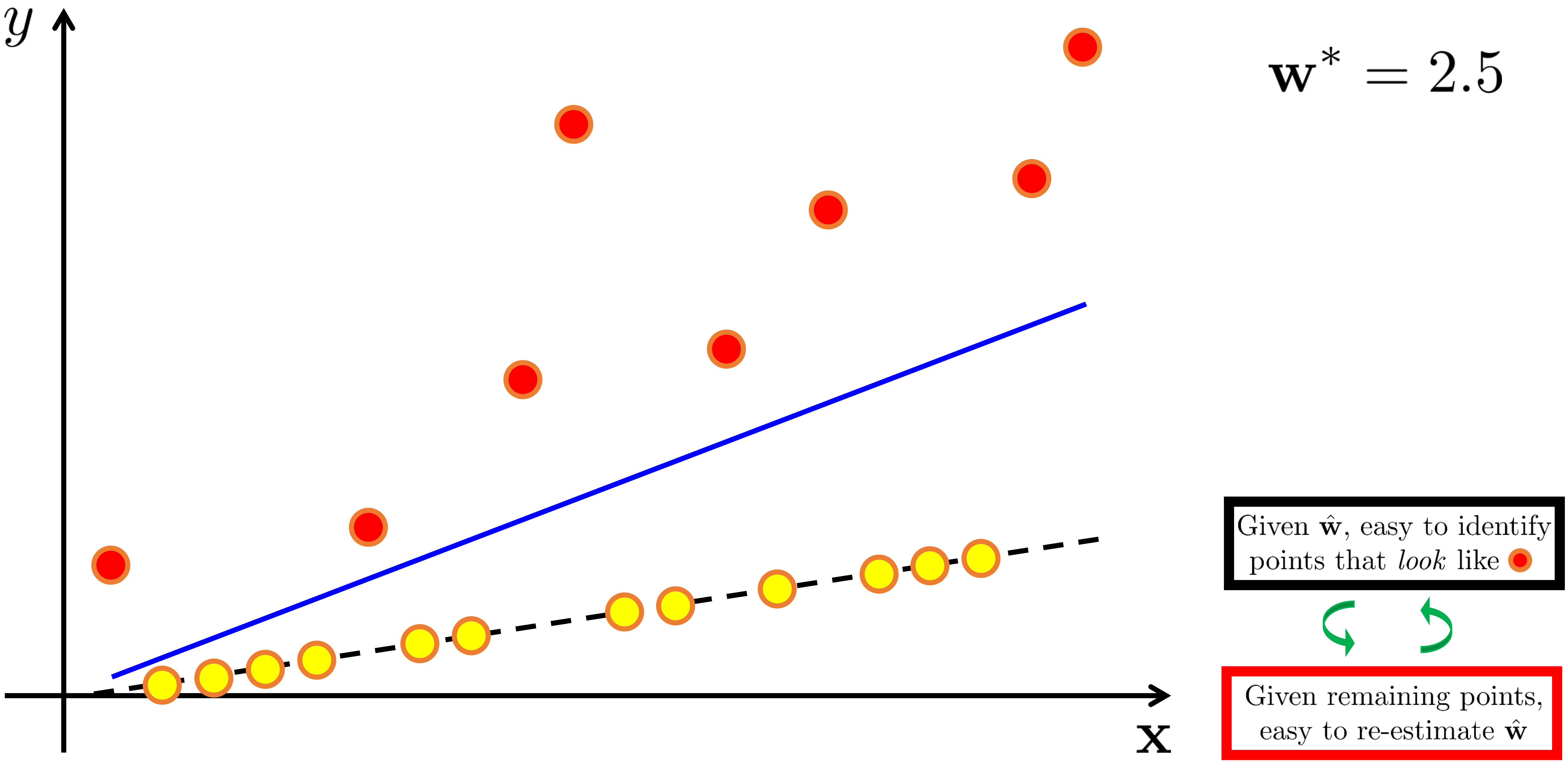


$$\mathbf{w}^* = 2.5$$

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!

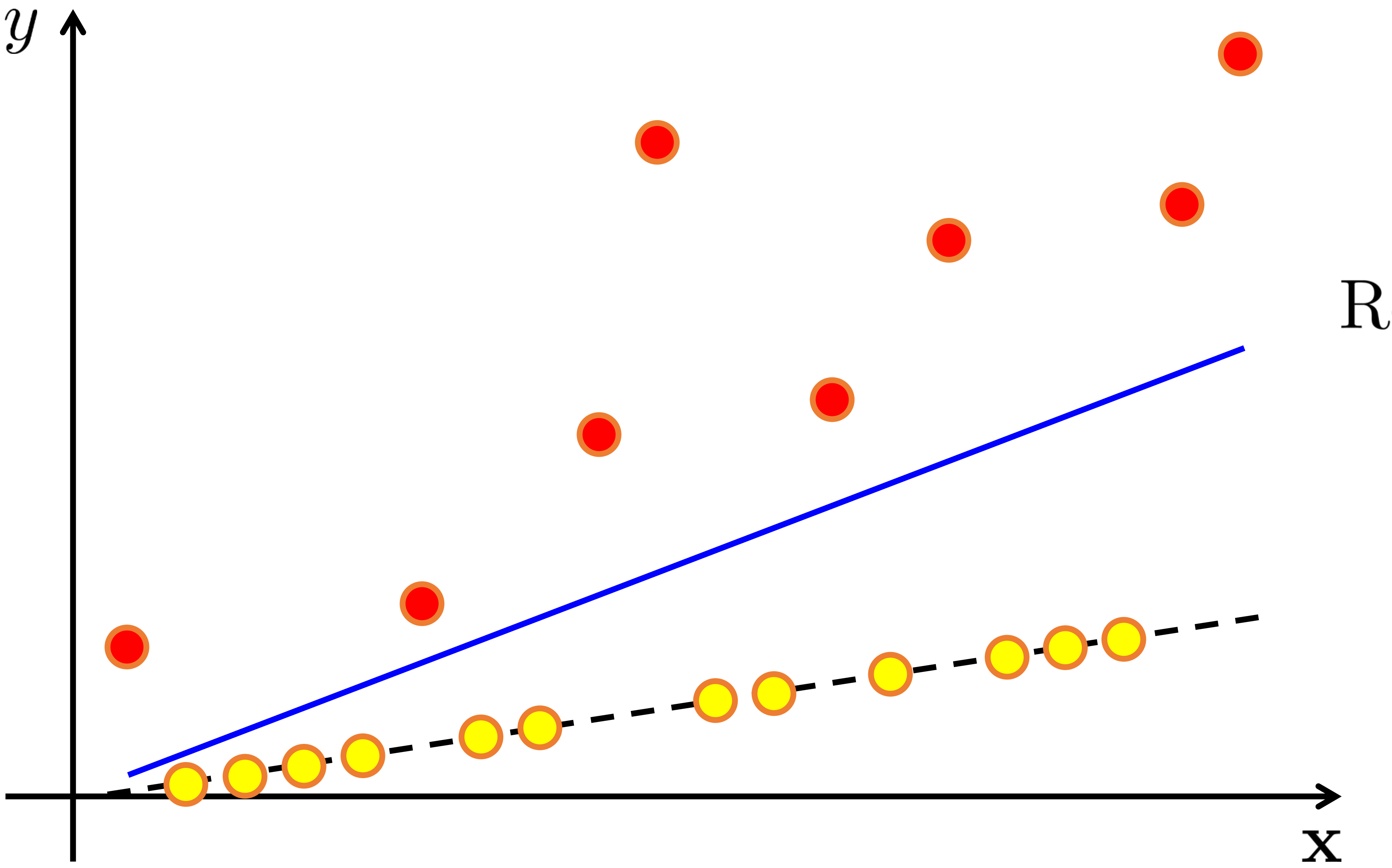


$\mathbf{w}^* = 2.5$

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

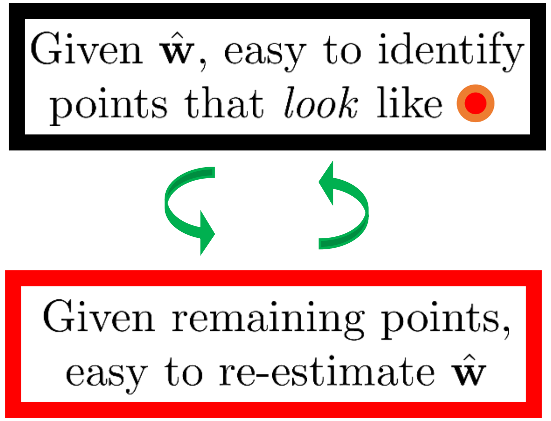
TORRENT in Action!



$\mathbf{w}^* = 2.5$

Residual: 11.7

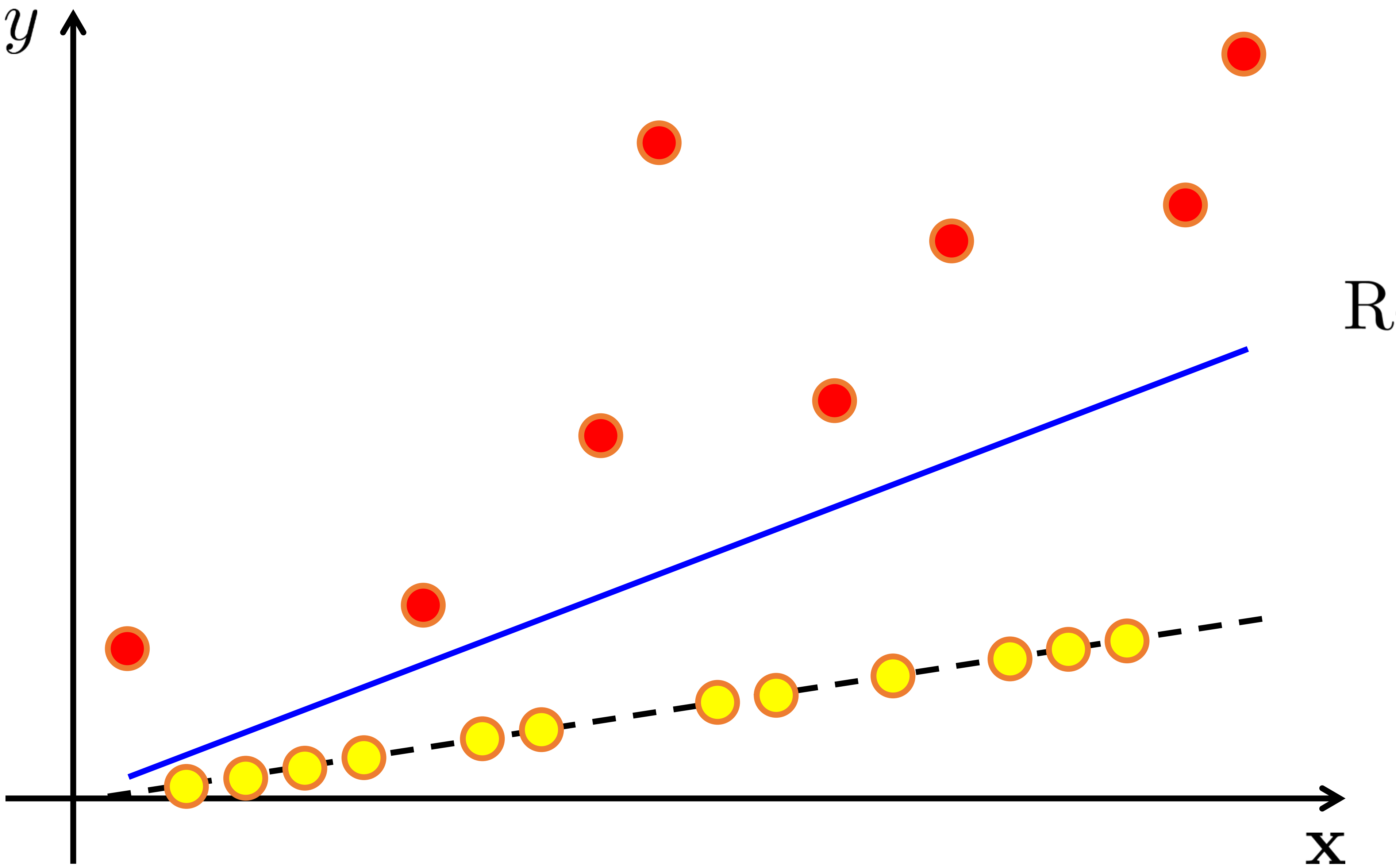
$\hat{\mathbf{w}}$: 6.3



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

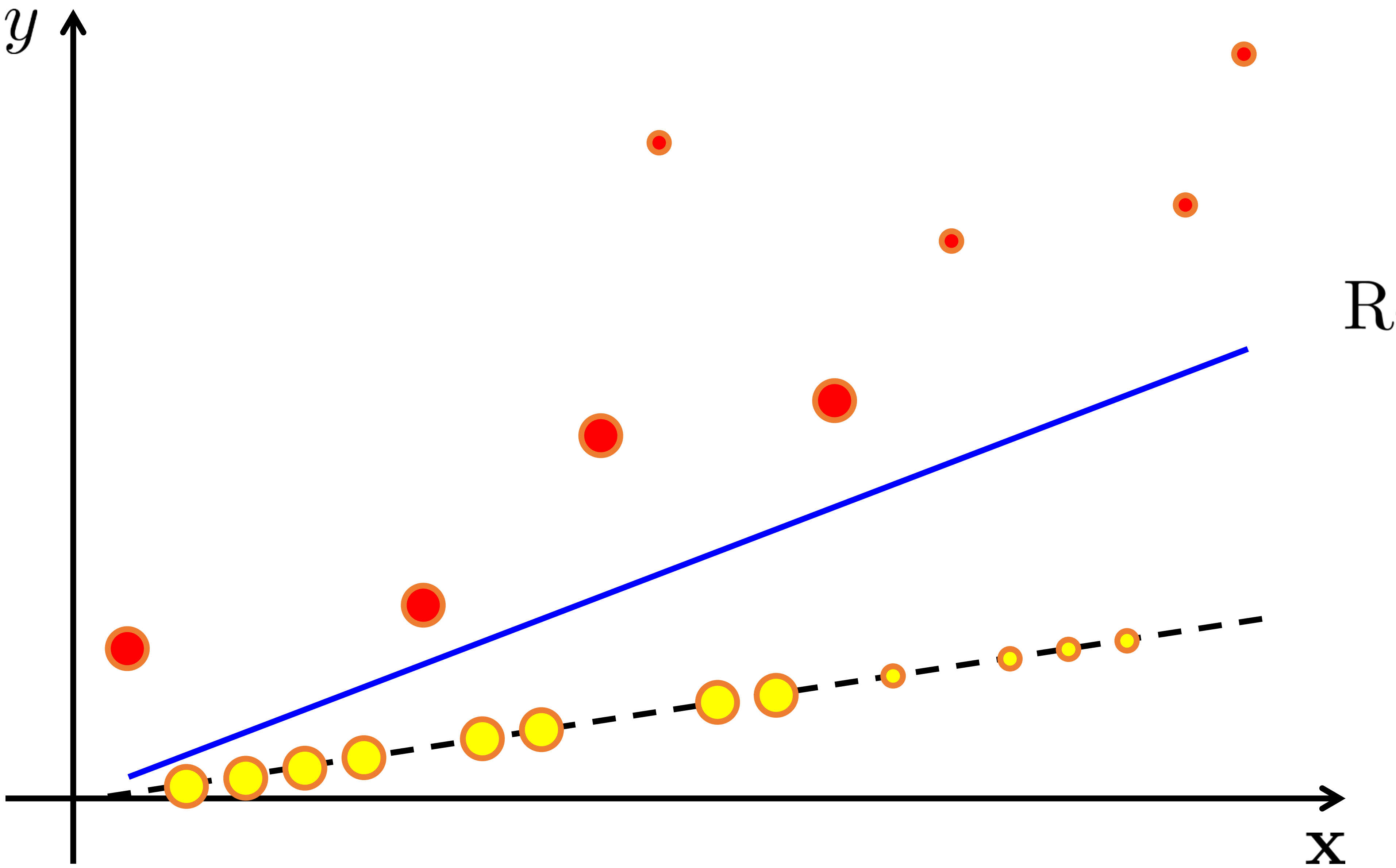
Residual: 11.7

$\hat{\mathbf{w}}$: 6.3

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

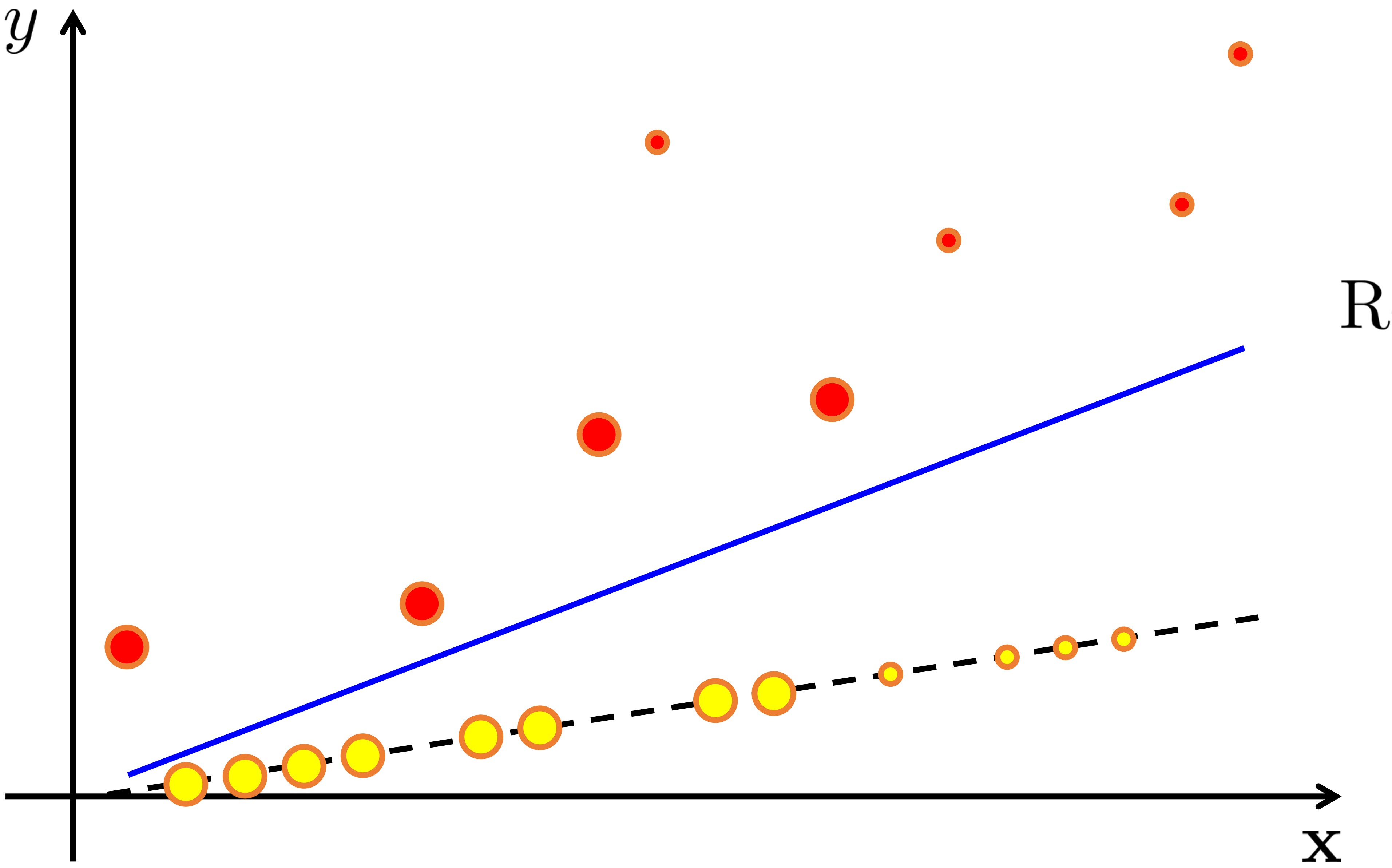
Residual: 11.7

$\hat{\mathbf{w}}$: 6.3

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

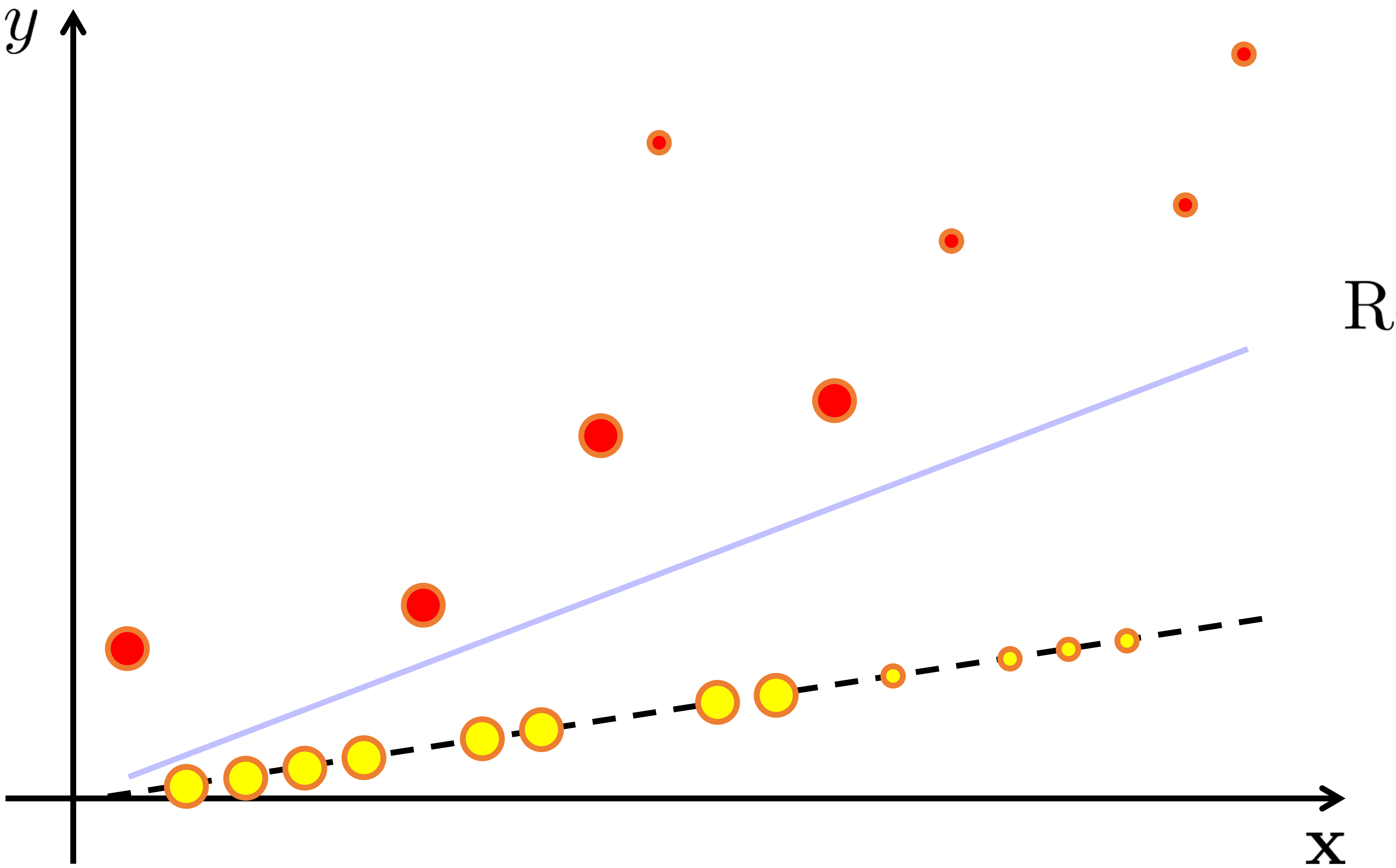
Residual: 11.7

$\hat{\mathbf{w}}$: 6.3

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

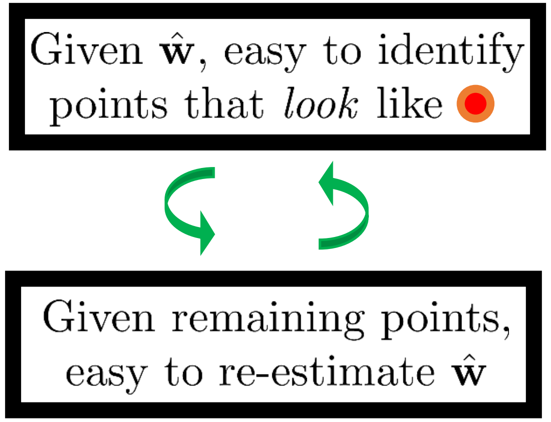
TORRENT in Action!



$\mathbf{w}^* = 2.5$

Residual: 11.7

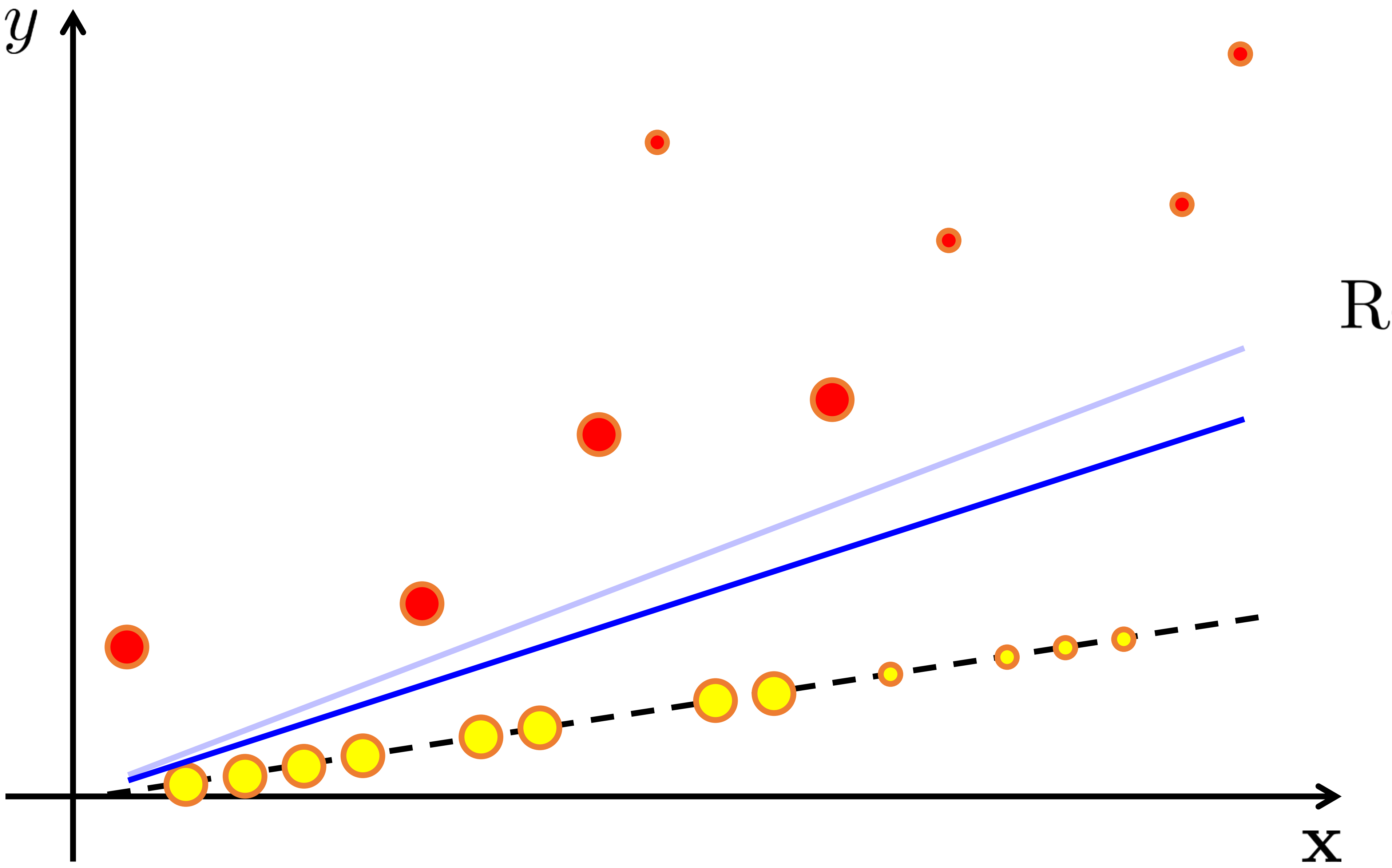
$\hat{\mathbf{w}}$: 6.3



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

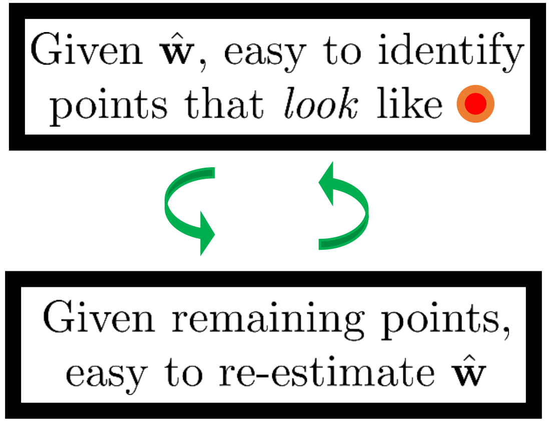
TORRENT in Action!



$\mathbf{w}^* = 2.5$

Residual: 11.7

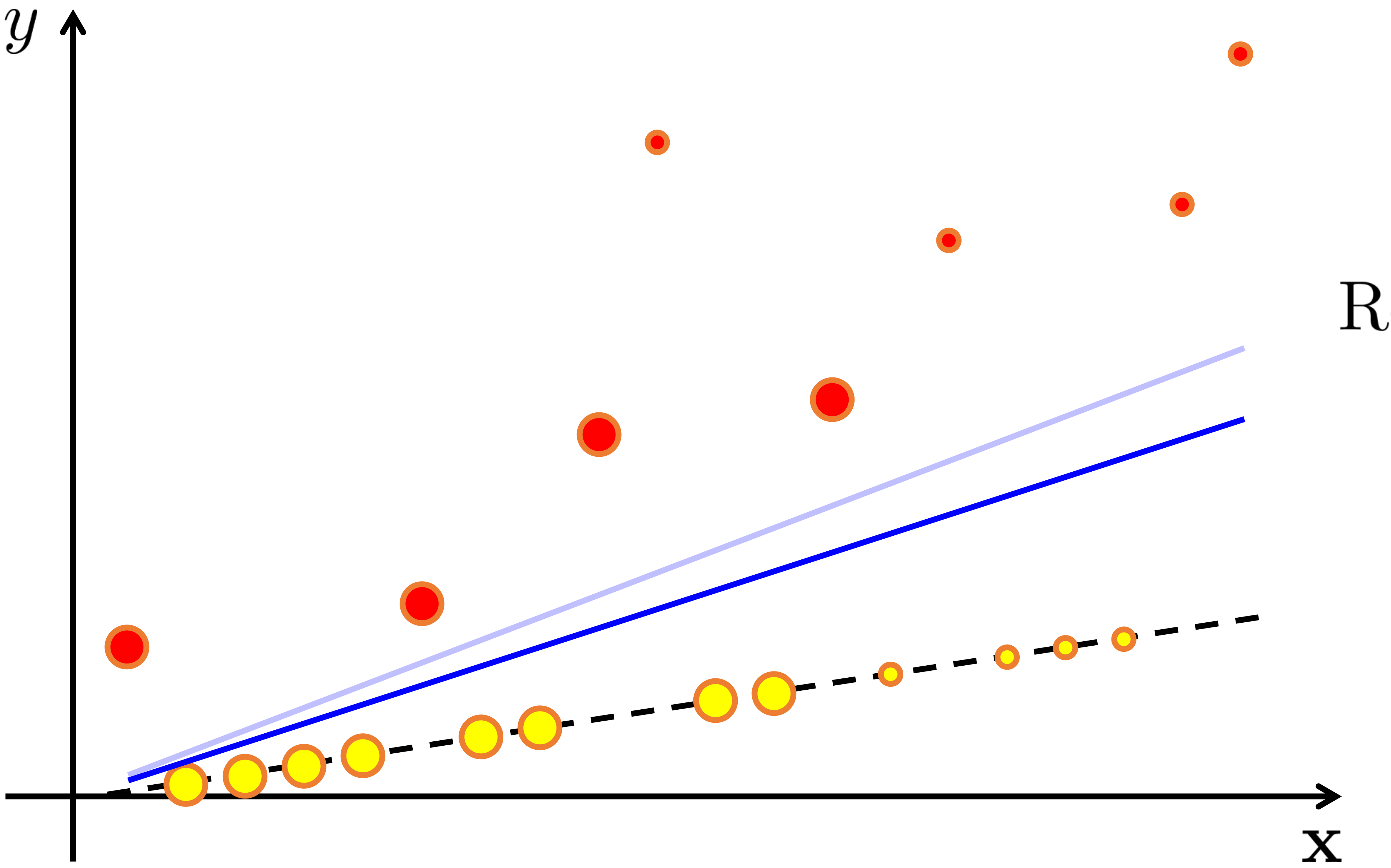
$\hat{\mathbf{w}}$: 6.3



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

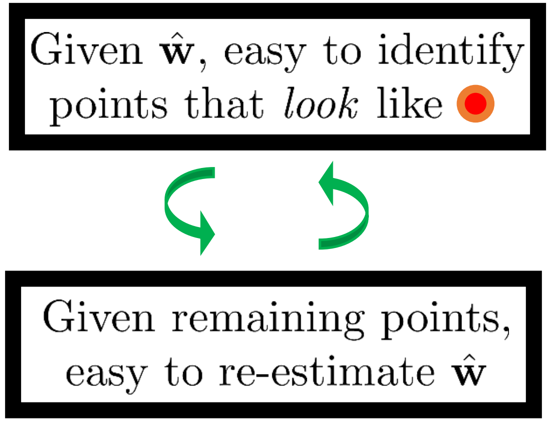
TORRENT in Action!



$$\mathbf{w}^* = 2.5$$

Residual: 5.05

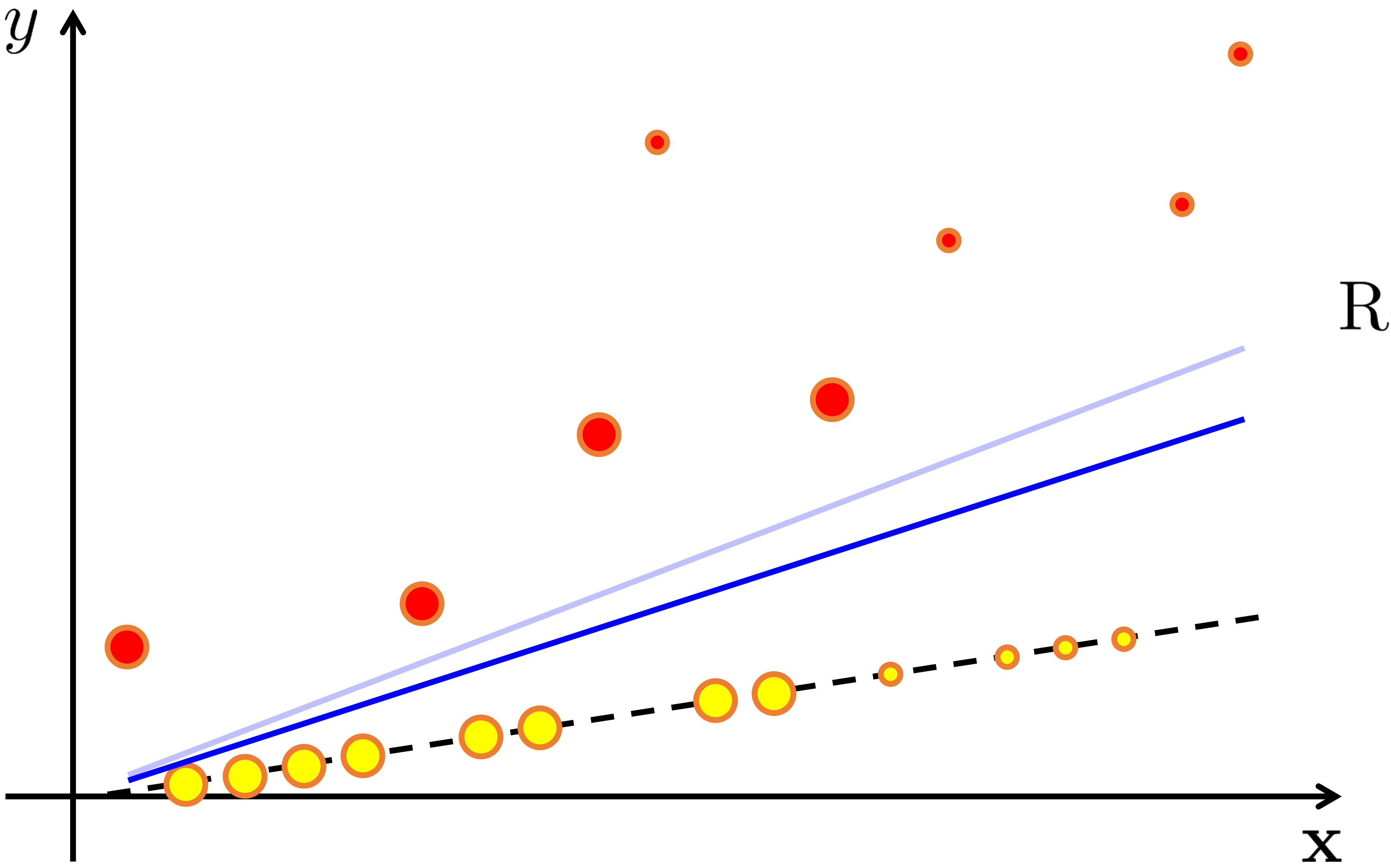
$\hat{\mathbf{w}}$: 5.3



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

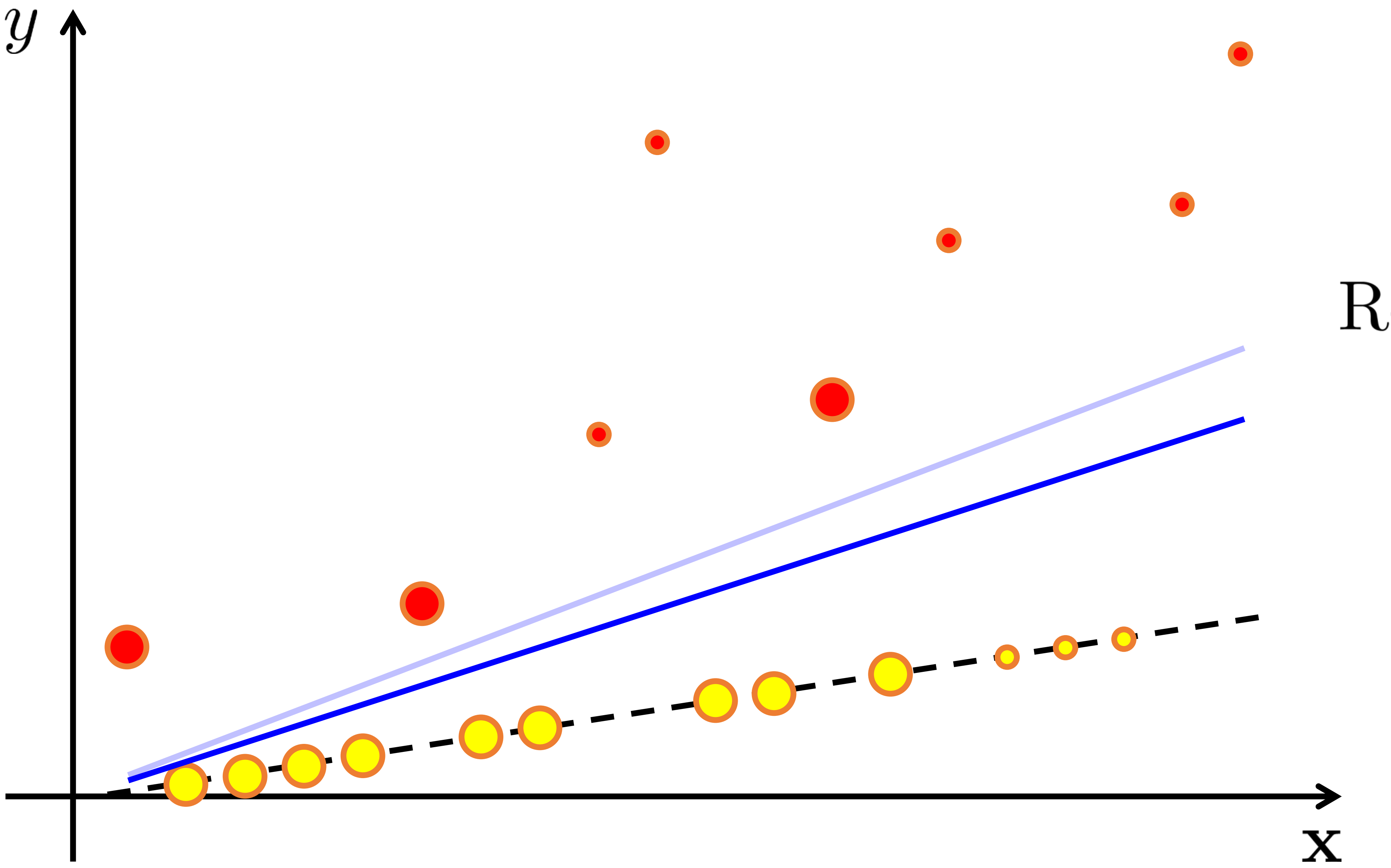
Residual: 5.05

$\hat{\mathbf{w}}$: 5.3

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

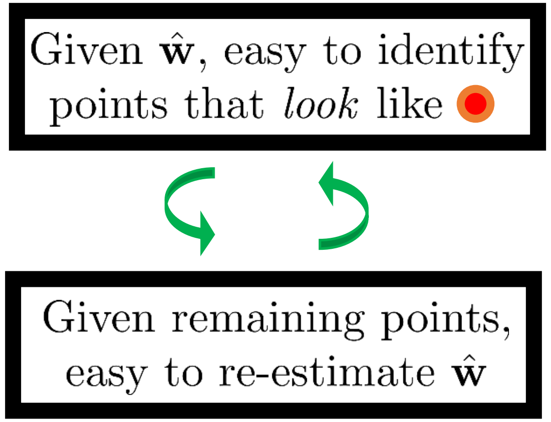
TORRENT in Action!



$\mathbf{w}^* = 2.5$

Residual: 5.05

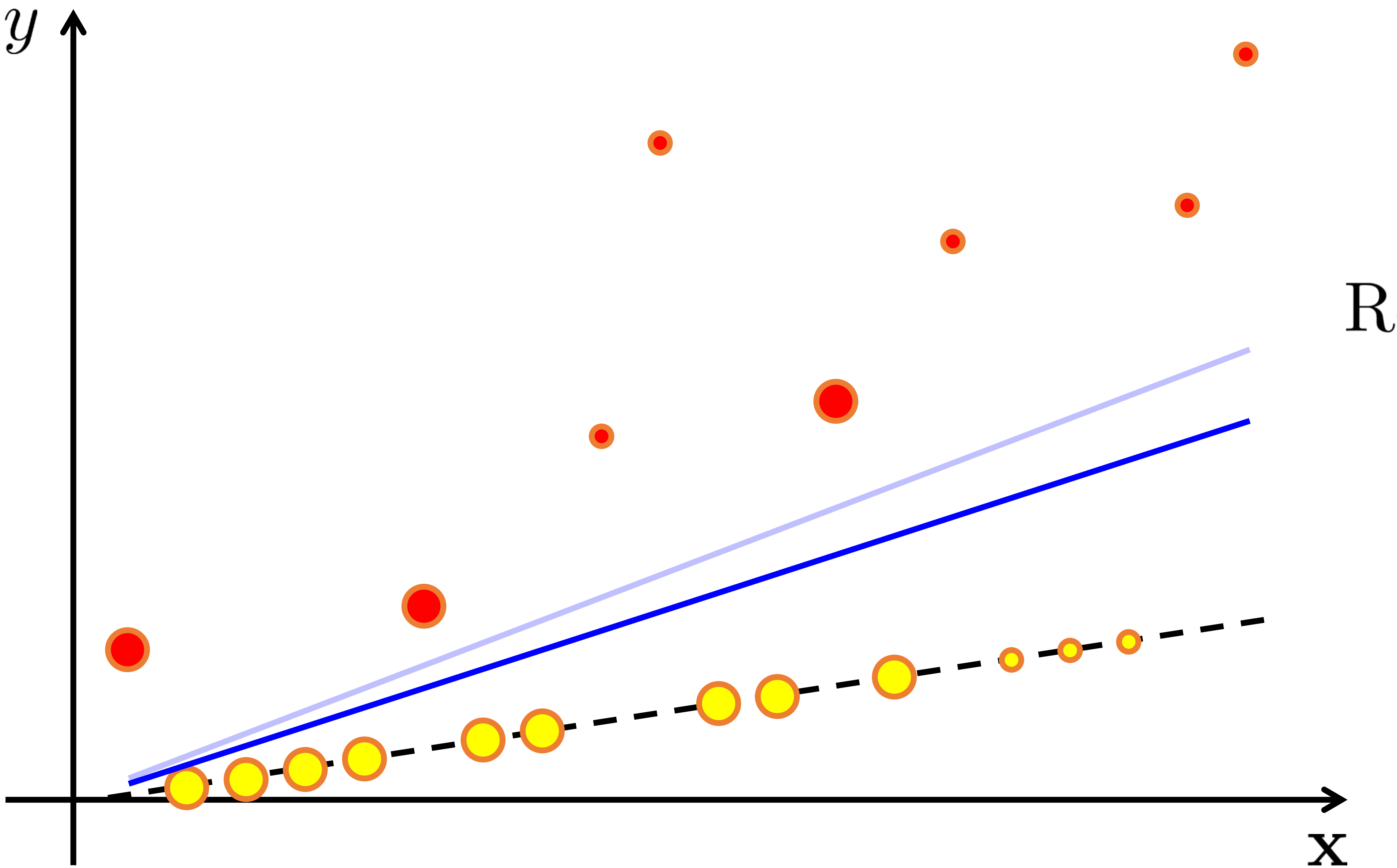
$\hat{\mathbf{w}}$: 5.3



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

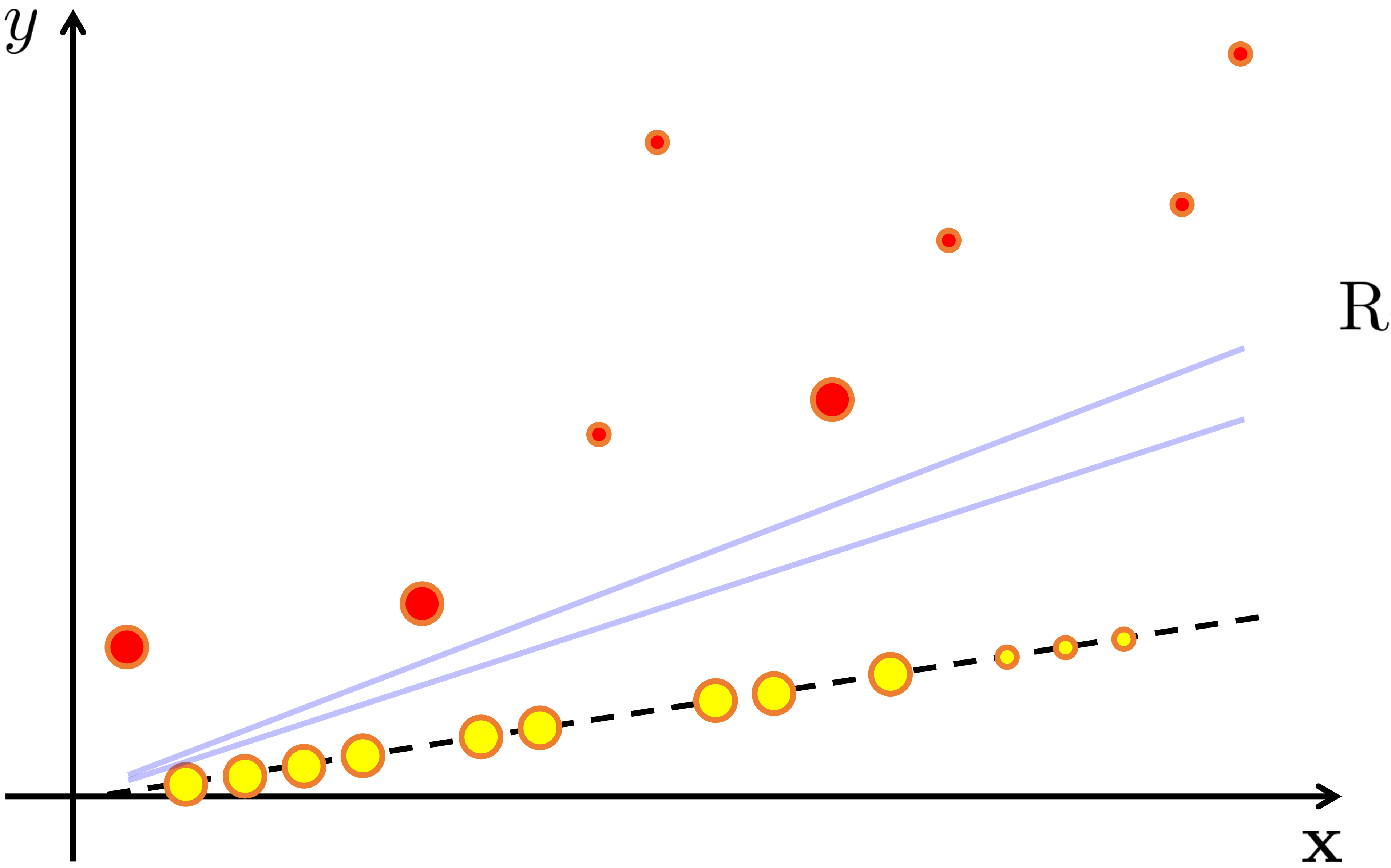
Residual: 5.05

$\hat{\mathbf{w}}$: 5.3

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

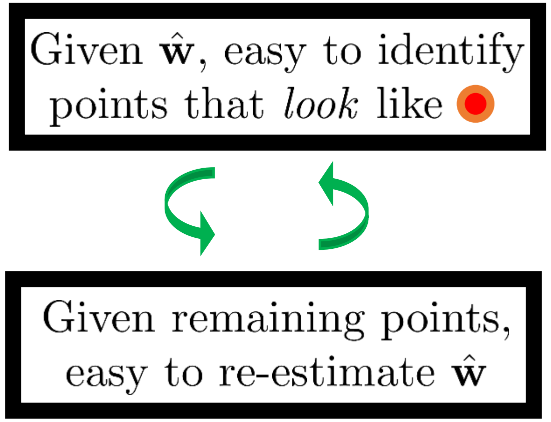
TORRENT in Action!



$\mathbf{w}^* = 2.5$

Residual: 5.05

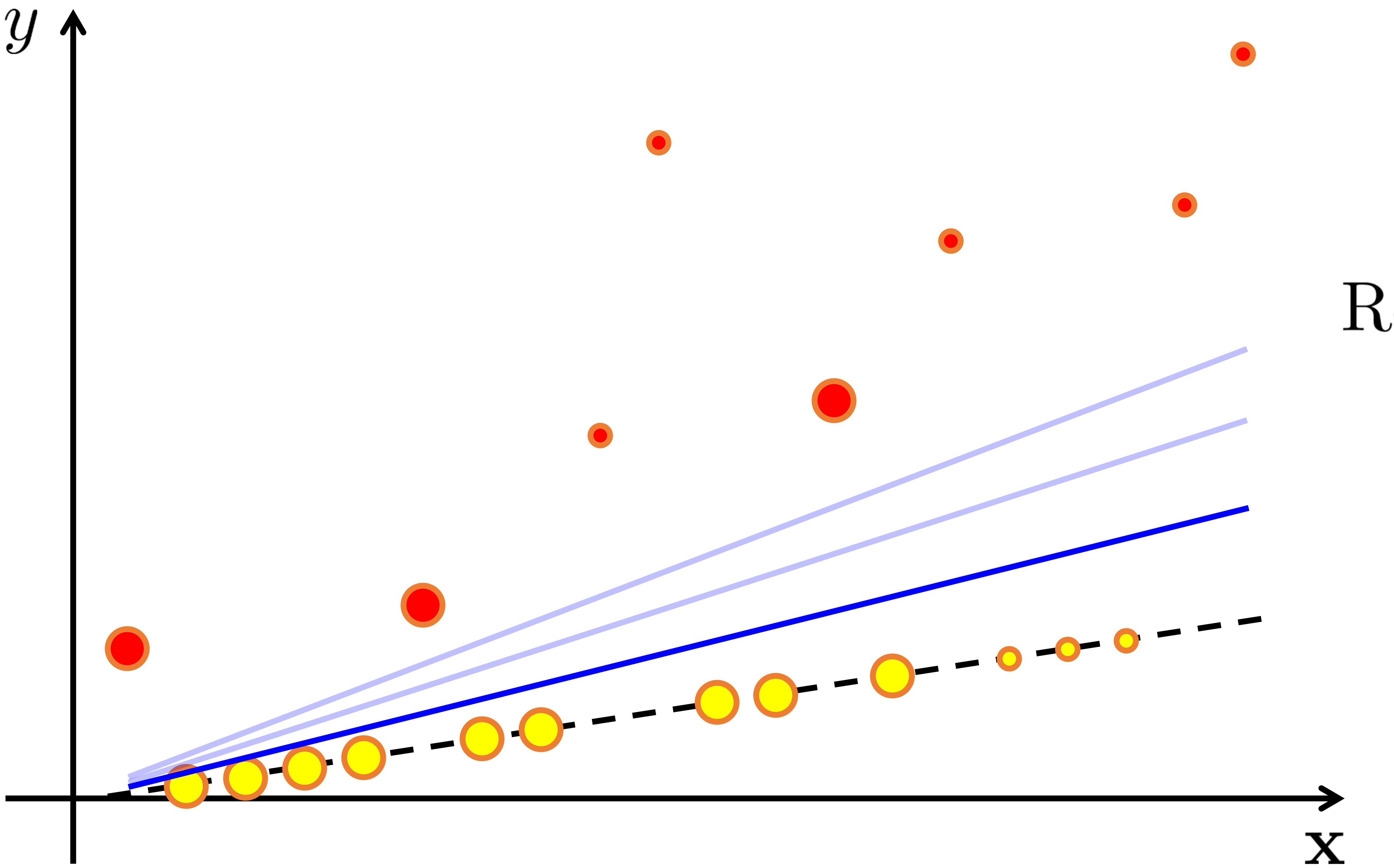
$\hat{\mathbf{w}}$: 5.3



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

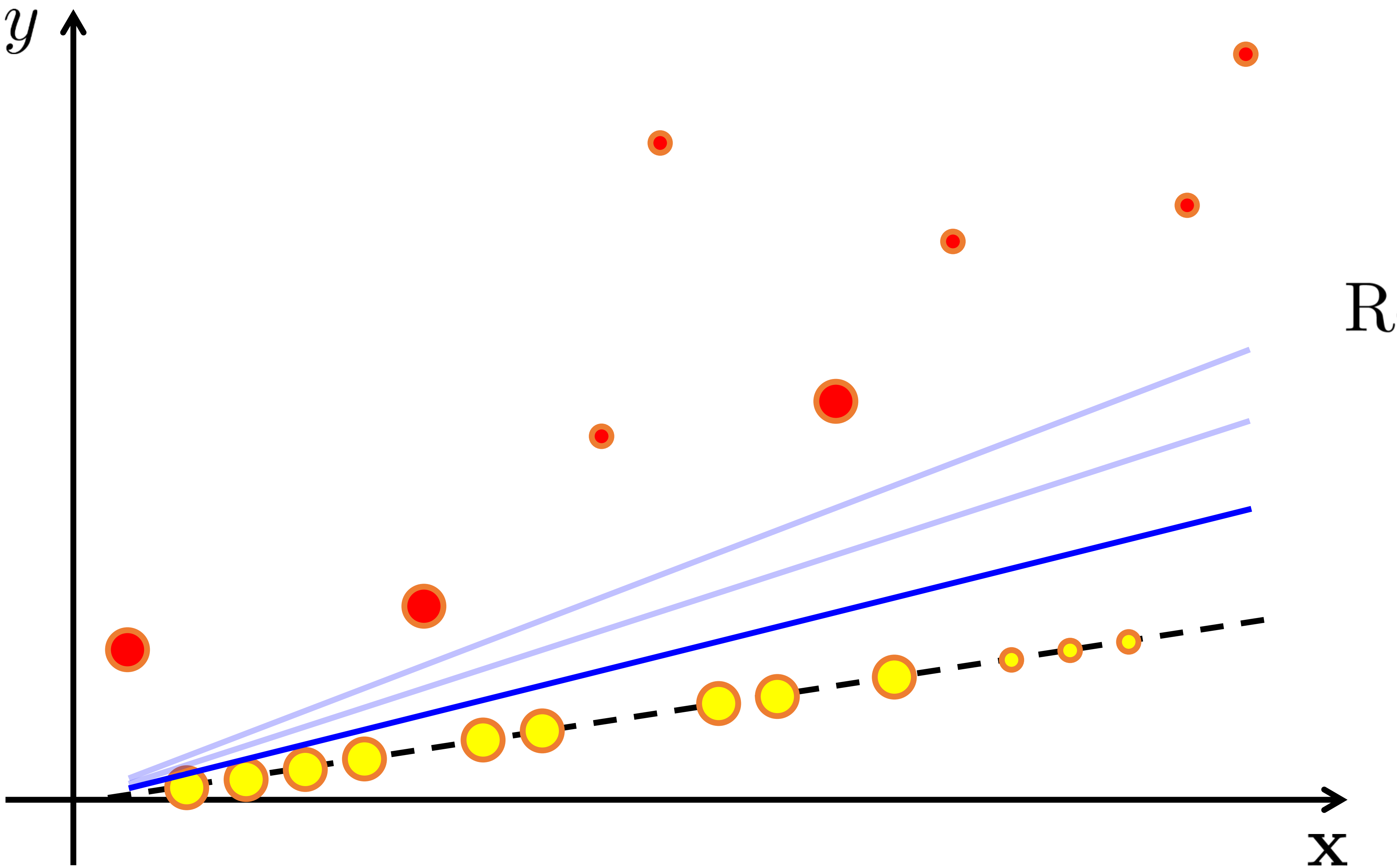
Residual: 5.05

$\hat{\mathbf{w}}$: 5.3

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

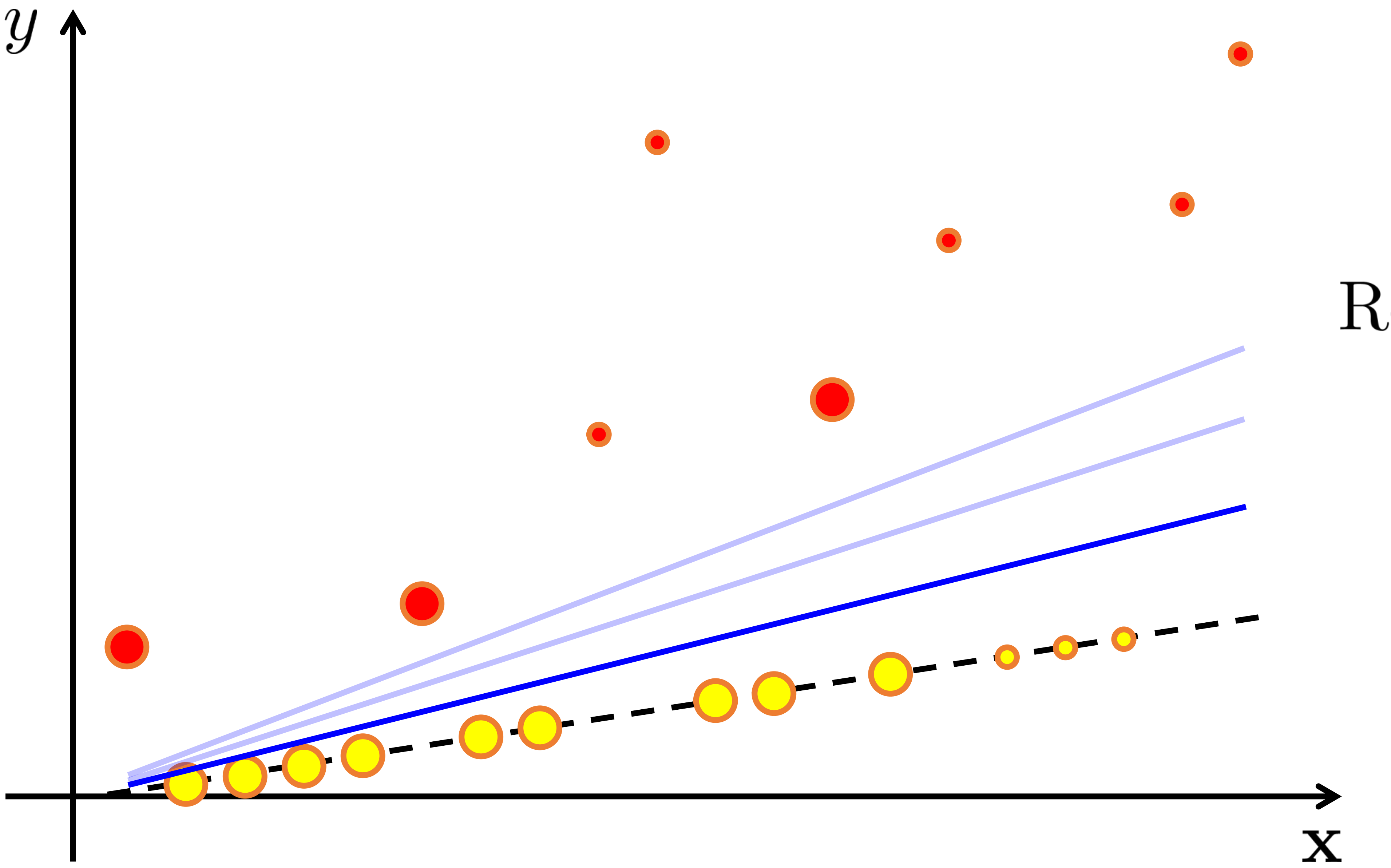
Residual: 4.33

$\hat{\mathbf{w}}$: 4.1

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

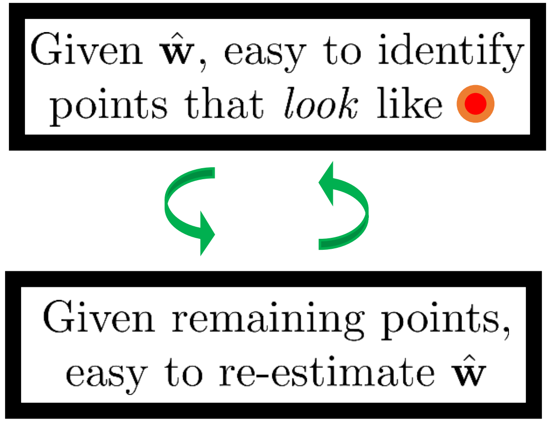
TORRENT in Action!



$\mathbf{w}^* = 2.5$

Residual: 4.33

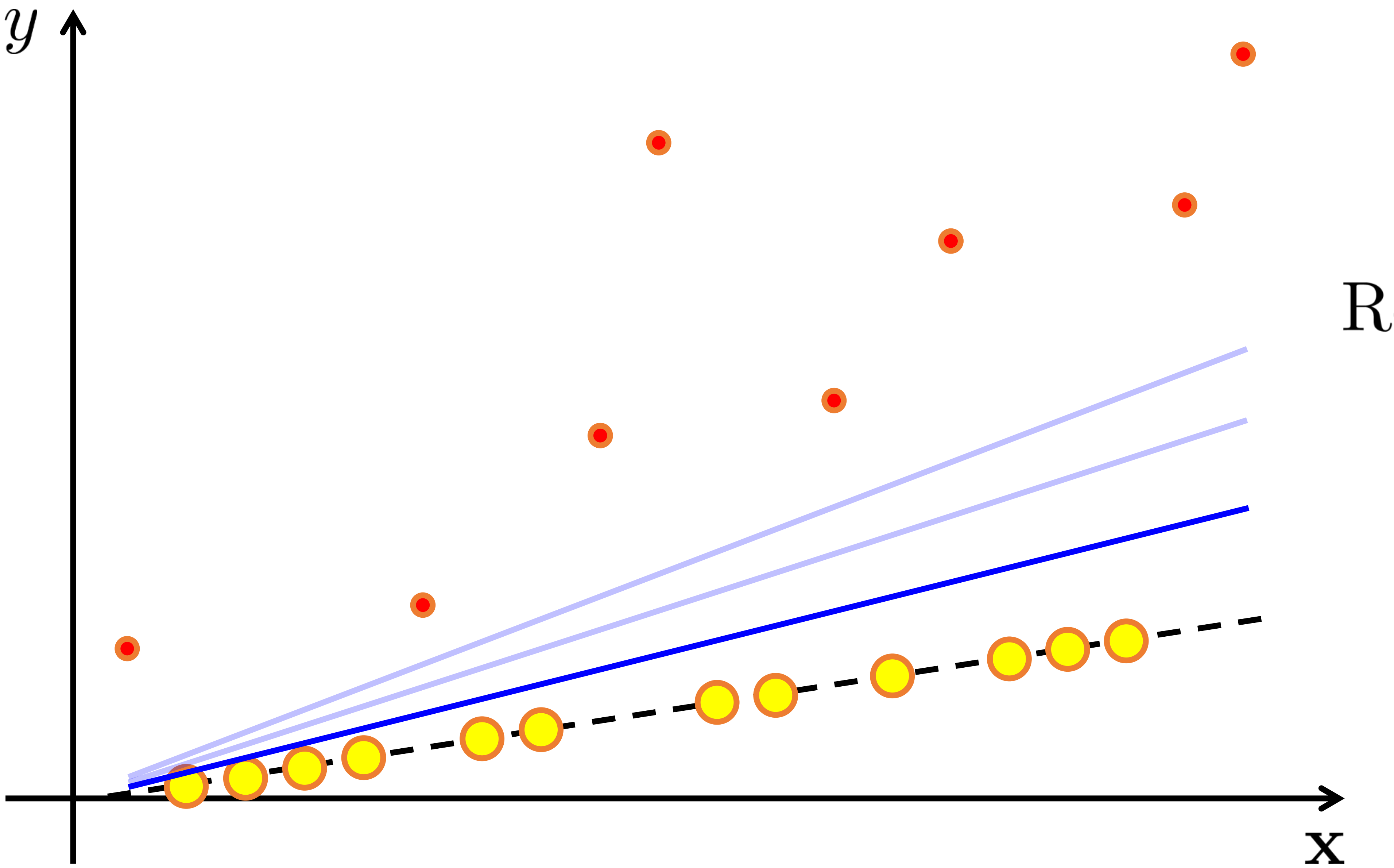
$\hat{\mathbf{w}}$: 4.1



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$$\mathbf{w}^* = 2.5$$

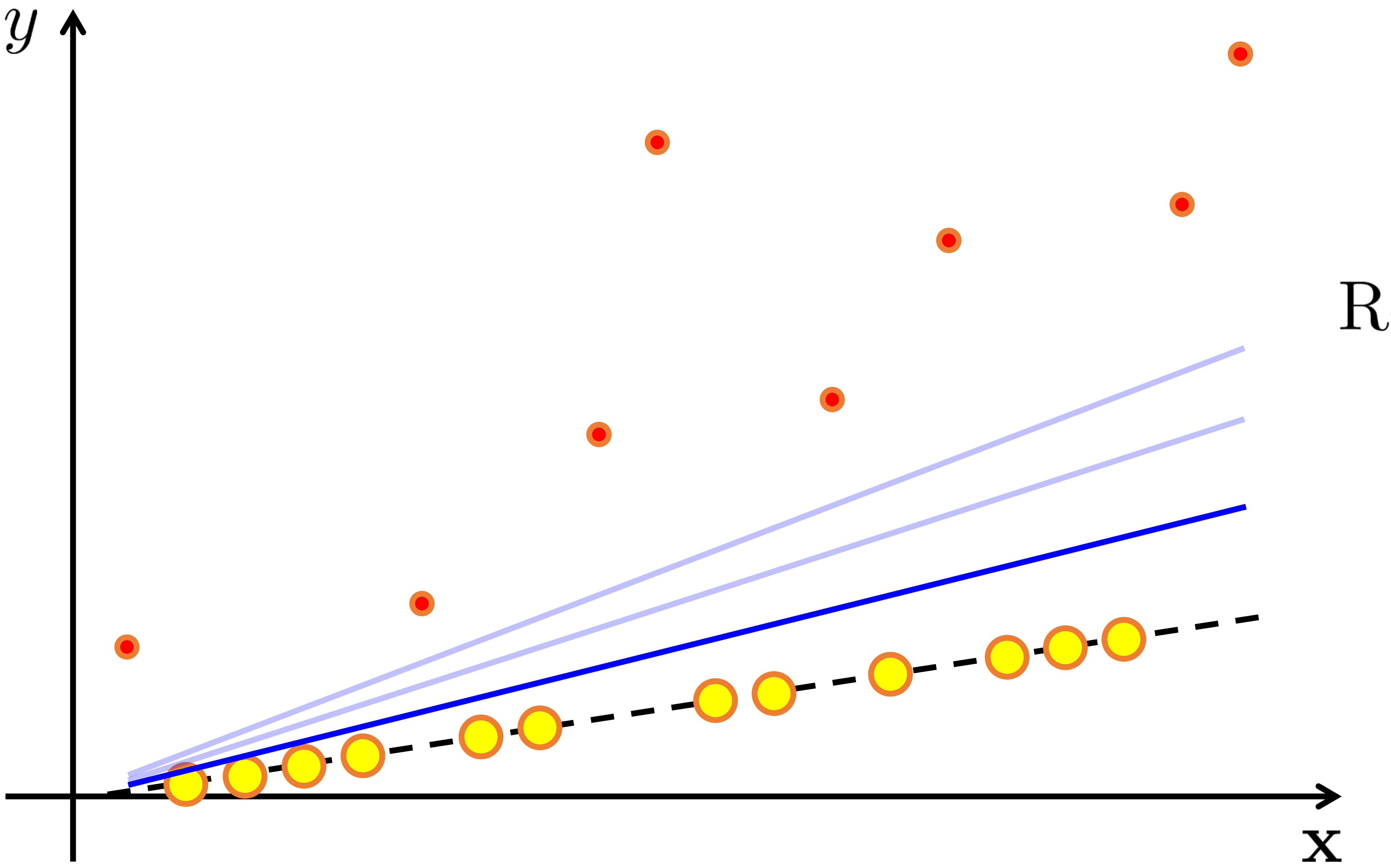
$$\text{Residual: } 4.33$$

$$\hat{\mathbf{w}}\text{: } 4.1$$

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

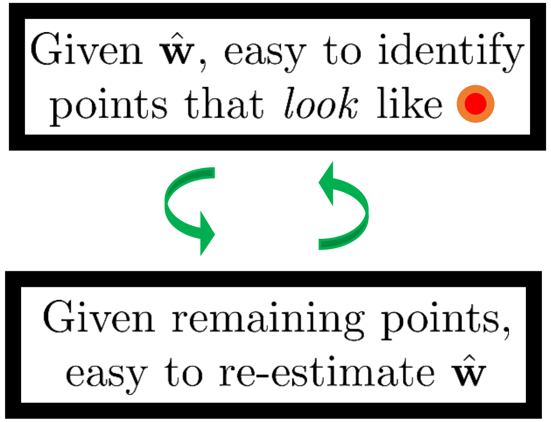
TORRENT in Action!



$\mathbf{w}^* = 2.5$

Residual: 4.33

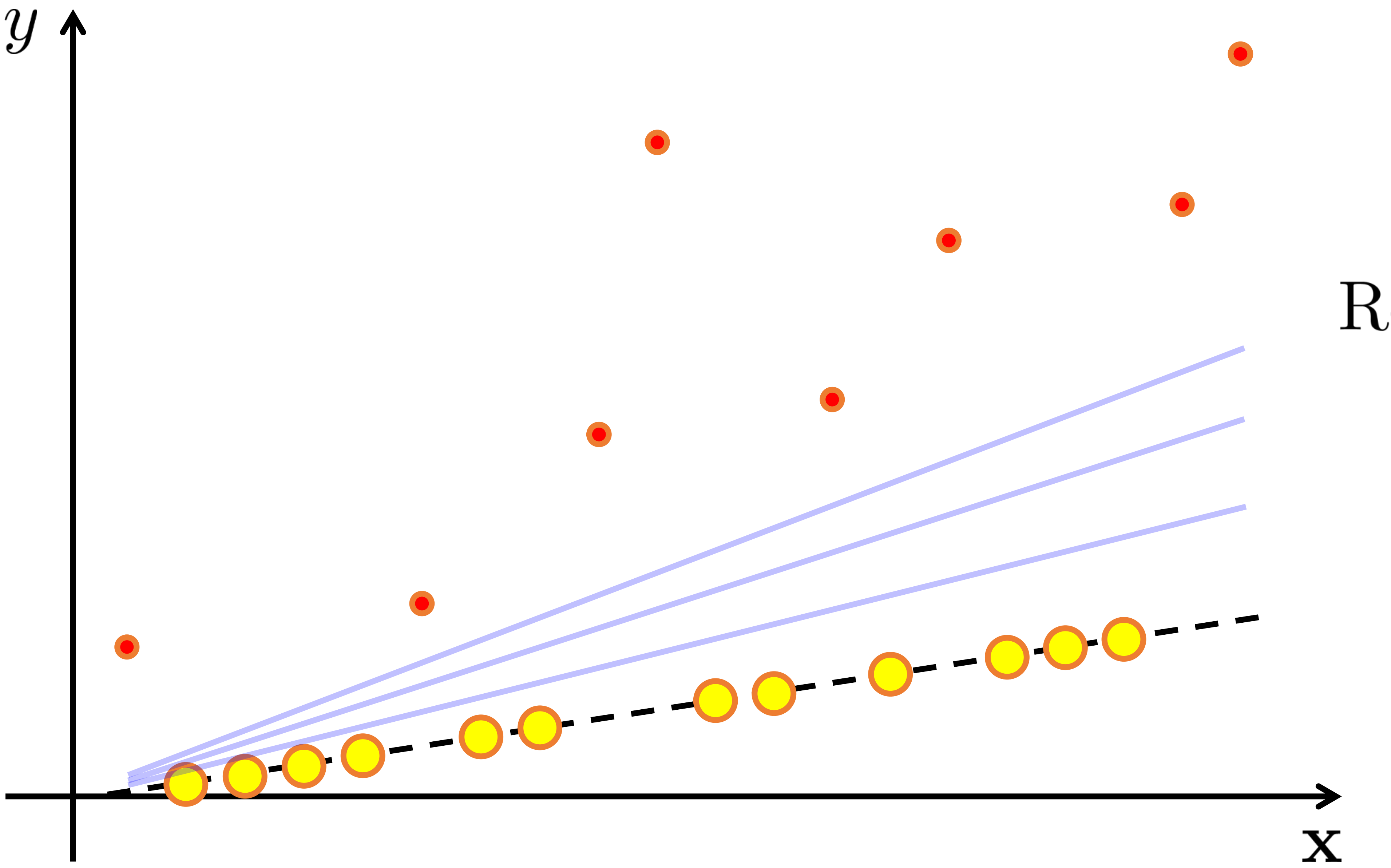
$\hat{\mathbf{w}}$: 4.1



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

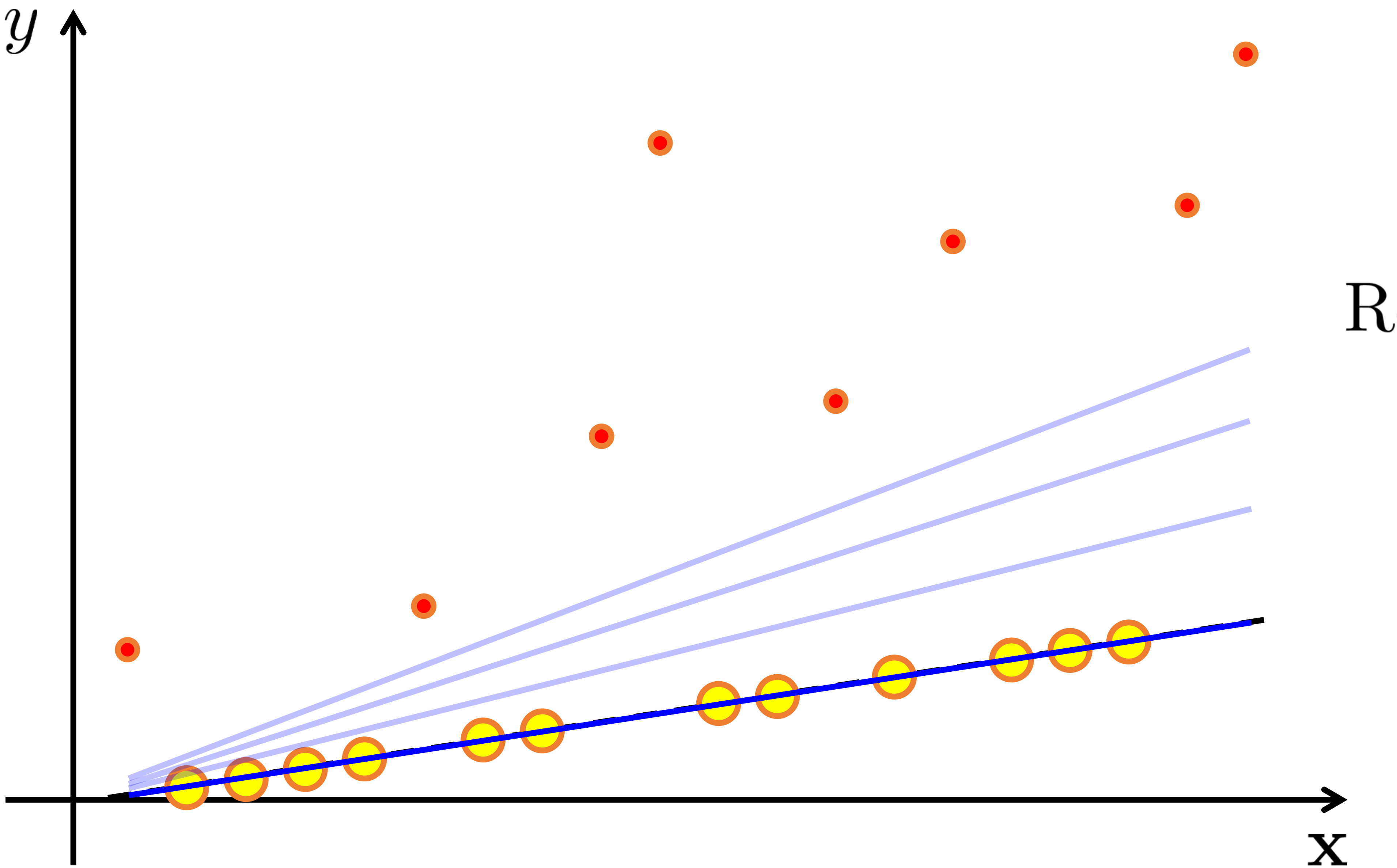
Residual: 4.33

$\hat{\mathbf{w}}$: 4.1

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

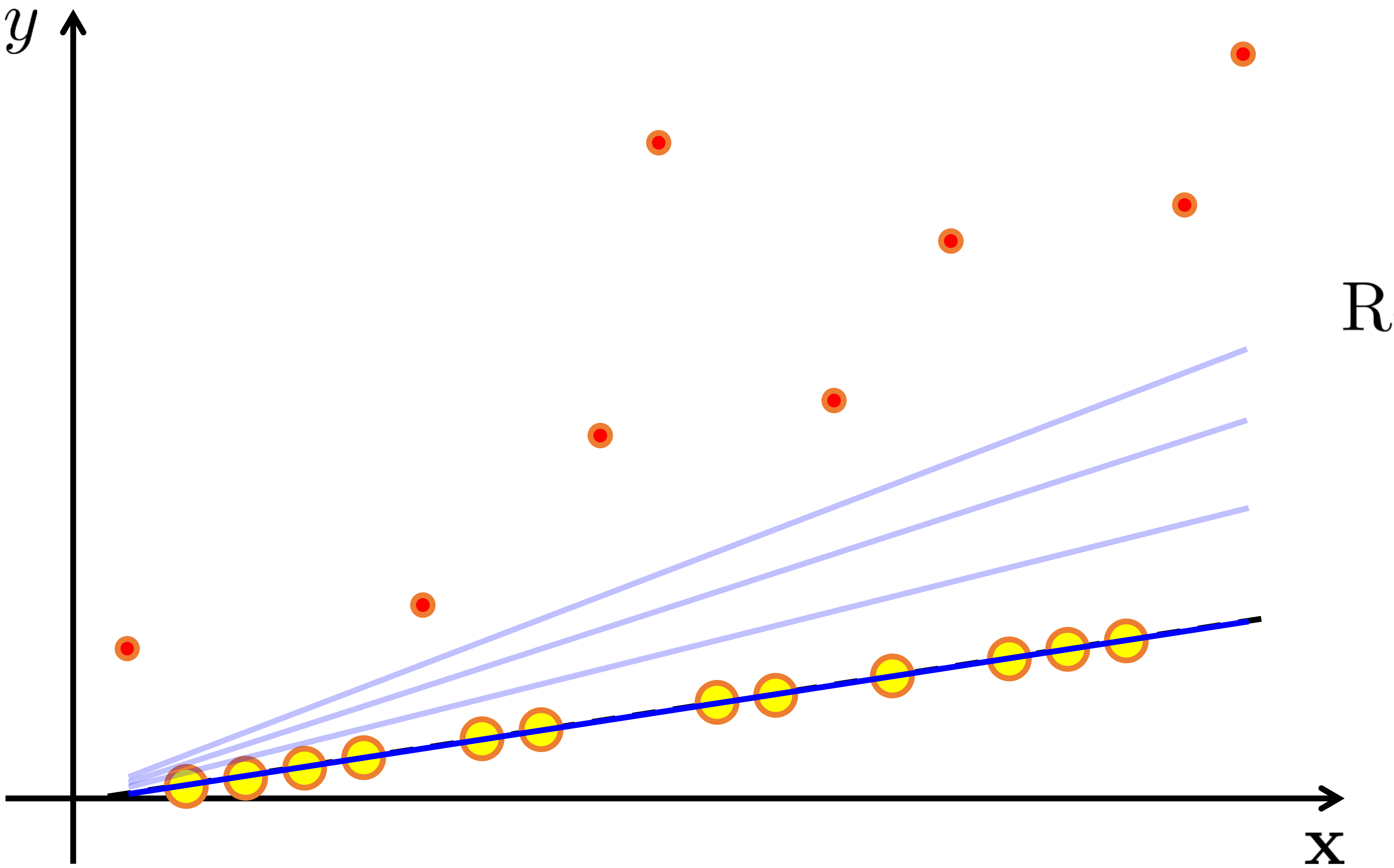
Residual: 4.33

$\hat{\mathbf{w}}$: 4.1

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$\mathbf{w}^* = 2.5$

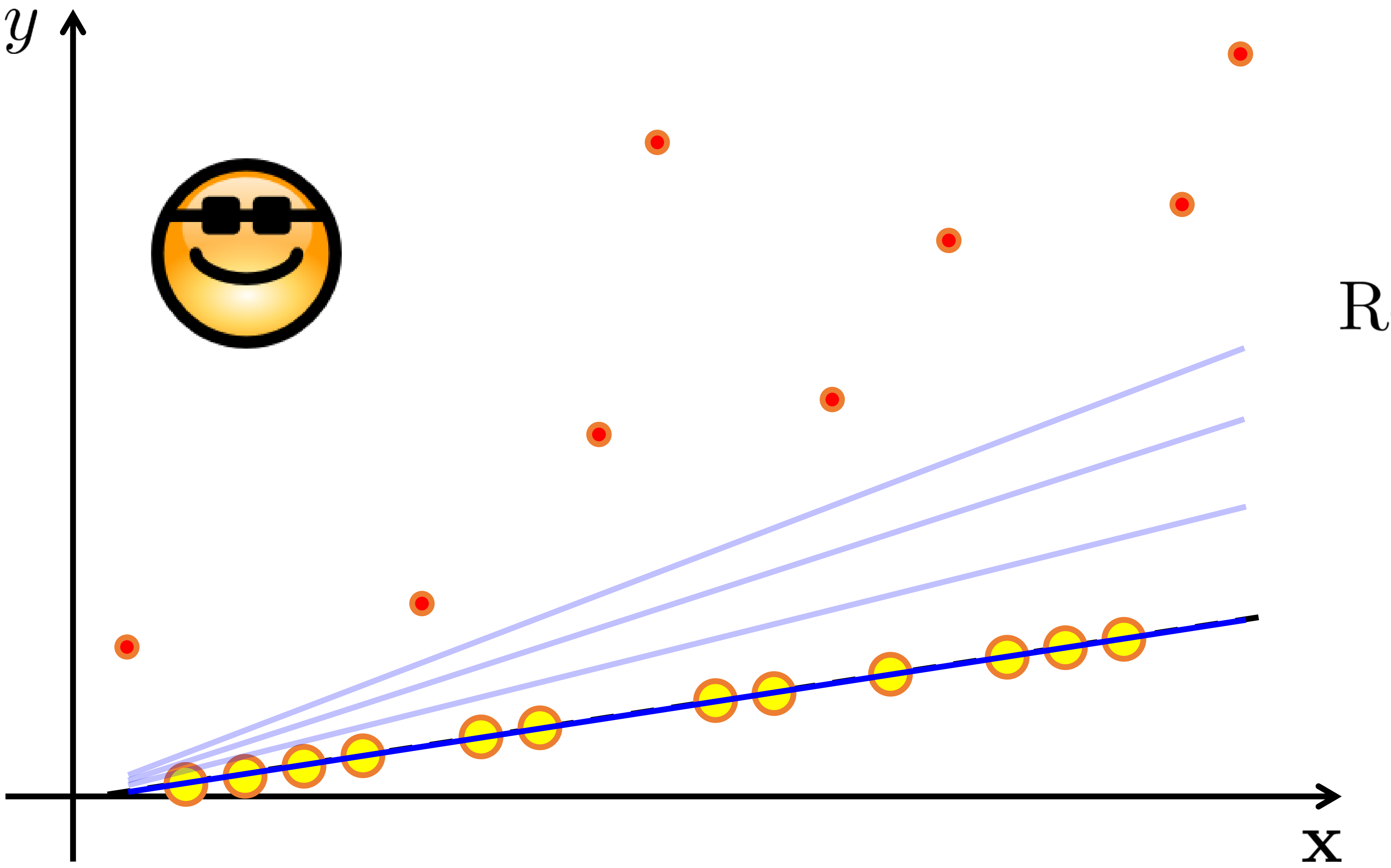
Residual: 0

$\hat{\mathbf{w}}$: 2.5

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

TORRENT in Action!



$$\mathbf{w}^* = 2.5$$

$$\text{Residual: } 0$$

$$\hat{\mathbf{w}}\text{: } 2.5$$

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to re-estimate $\hat{\mathbf{w}}$

Alt-Min in Theory

Recovery Guarantees

Robust against adaptive adversaries

has access to data $\mathbf{x}_i$, gold model $\mathbf{w}^*$, and noise e_i

Requirement:

Data $\mathbf{X}$ needs to satisfy some "nice" properties

Enough data needs to be present $n = \Omega(p \lg p)$

Guarantees:

TORRENT will recover the gold model if $\alpha \leq \frac{1}{60}$ i.e. $k \leq \frac{n}{60}$

Alt-Min in Theory

Recovery Guarantees

Robust against adaptive adversaries

has access to data $\mathbf{x}_i$, gold model $\mathbf{w}^*$, and noise e_i

Requirement:

Data $\mathbf{X}$ needs to satisfy some “nice” properties

Enough data needs to be present $n = \Omega(p \lg p)$

Guarantees:

TORRENT will recover the gold model if $\alpha \leq \frac{1}{60}$ i.e. $k \leq \frac{n}{60}$

Alt-Min in Theory

Convergence Rates

Linear rate of convergence

Suppose each alternation ≡ one step

After $T = \log \frac{1}{\epsilon}$ time steps

$$\left\| \mathbf{w}^T - \mathbf{w}^* \right\|_2 \leq \epsilon$$

Invariant: at time t, “active set” $\mathcal{A}^t$ s.t

$$\|b_{\mathcal{A}^t}\|_2 \leq \frac{1}{2} \cdot \|b_{\mathcal{A}^{t-1}}\|_2$$

Alt-Min in Theory

Convergence Rates

Linear rate of convergence

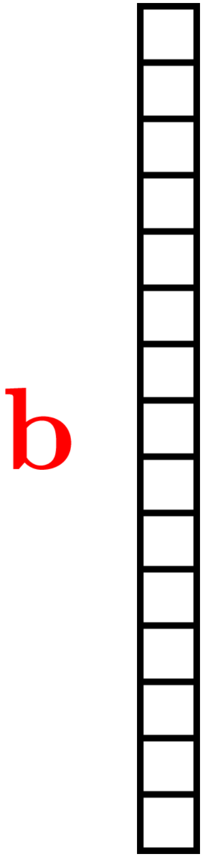
Suppose each alternation ≡ one step

After $T = \log\frac{1}{\epsilon}$ time steps

$$\left\|\mathbf{w}^T - \mathbf{w}^*\right\|_2 \leq \epsilon$$

Invariant: at time t, “active set” $\mathcal{A}^t$ s.t

$$\|b_{\mathcal{A}^t}\|_2 \leq \frac{1}{2} \cdot \|b_{\mathcal{A}^{t-1}}\|_2$$



Alt-Min in Theory

Convergence Rates

Linear rate of convergence

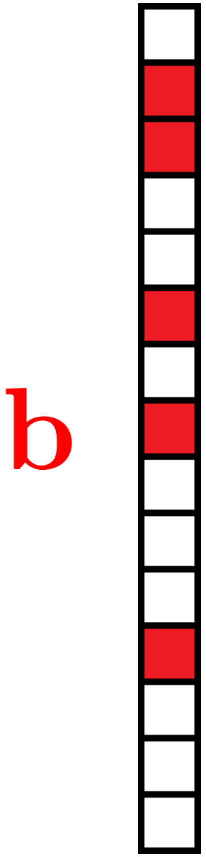
Suppose each alternation ≡ one step

After $T = \log\frac{1}{\epsilon}$ time steps

$$\left\|\mathbf{w}^T - \mathbf{w}^*\right\|_2 \leq \epsilon$$

Invariant: at time t, "active set" $\mathcal{A}^t$ s.t

$$\|b_{\mathcal{A}^t}\|_2 \leq \frac{1}{2} \cdot \|b_{\mathcal{A}^{t-1}}\|_2$$



Alt-Min in Theory

Convergence Rates

Linear rate of convergence

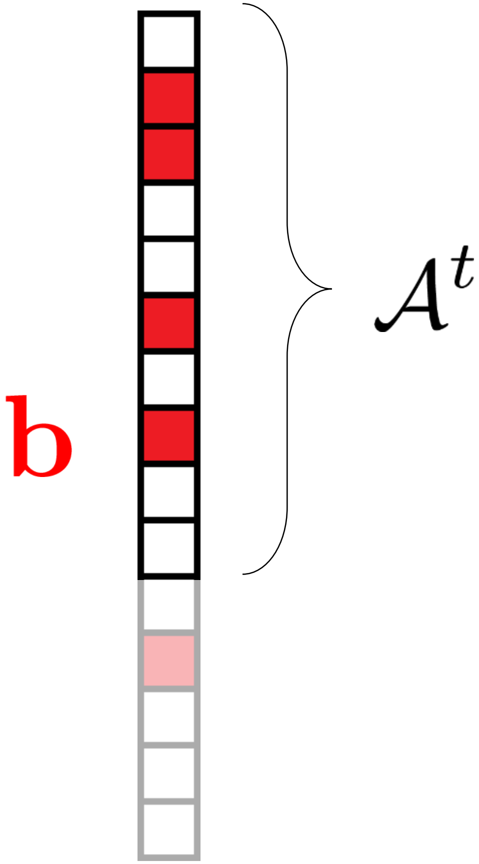
Suppose each alternation ≡ one step

After $T = \log\frac{1}{\epsilon}$ time steps

$$\left\|\mathbf{w}^T - \mathbf{w}^*\right\|_2 \leq \epsilon$$

Invariant: at time t, "active set" $\mathcal{A}^t$ s.t

$$\|b_{\mathcal{A}^t}\|_2 \leq \frac{1}{2} \cdot \|b_{\mathcal{A}^{t-1}}\|_2$$



Alt-Min in Theory

Convergence Rates

Linear rate of convergence

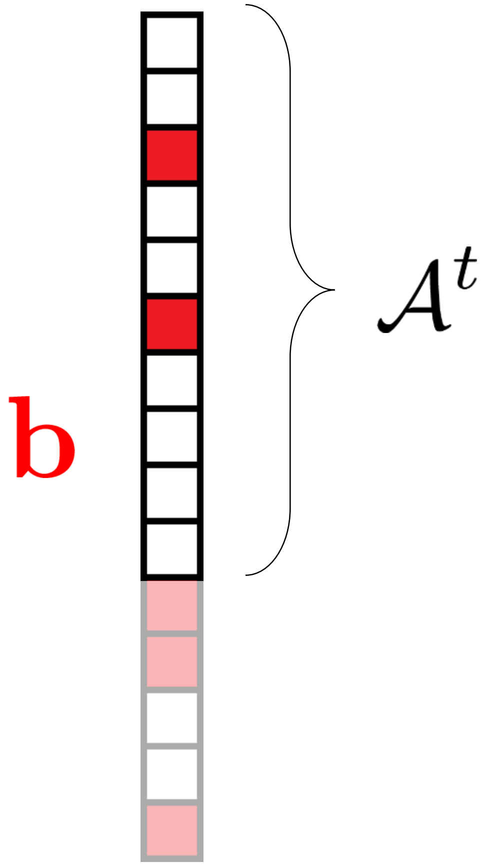
Suppose each alternation ≡ one step

After $T = \log\frac{1}{\epsilon}$ time steps

$$\left\|\mathbf{w}^T - \mathbf{w}^*\right\|_2 \leq \epsilon$$

Invariant: at time t, "active set" $\mathcal{A}^t$ s.t

$$\|b_{\mathcal{A}^t}\|_2 \leq \frac{1}{2} \cdot \|b_{\mathcal{A}^{t-1}}\|_2$$



Alt-Min in Theory

Convergence Rates

Linear rate of convergence

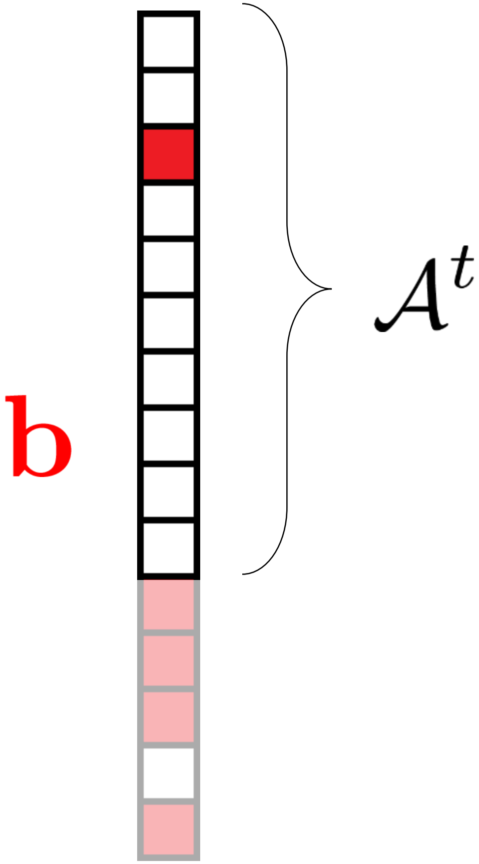
Suppose each alternation ≡ one step

After $T = \log\frac{1}{\epsilon}$ time steps

$$\left\|\mathbf{w}^T - \mathbf{w}^*\right\|_2 \leq \epsilon$$

Invariant: at time t, “active set” $\mathcal{A}^t$ s.t

$$\|b_{\mathcal{A}^t}\|_2 \leq \frac{1}{2} \cdot \|b_{\mathcal{A}^{t-1}}\|_2$$



Alt-Min in Theory

Convergence Rates

Linear rate of convergence

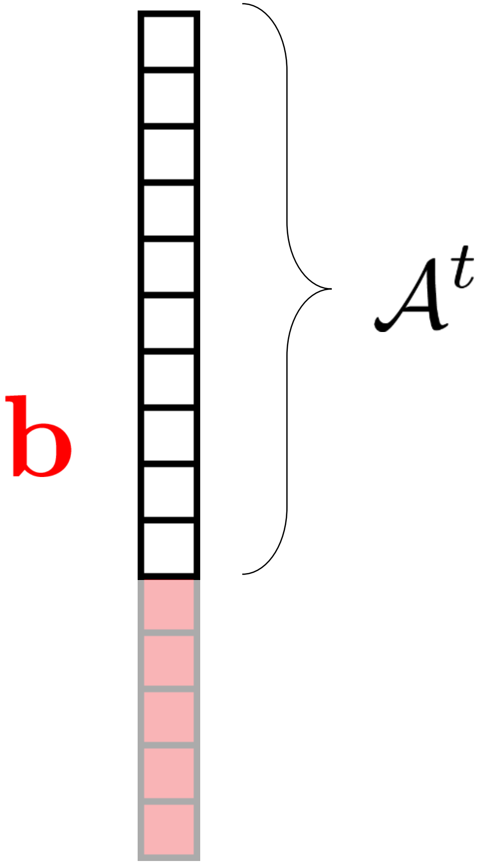
Suppose each alternation ≡ one step

After $T = \log\frac{1}{\epsilon}$ time steps

$$\left\|\mathbf{w}^T - \mathbf{w}^*\right\|_2 \leq \epsilon$$

Invariant: at time t, "active set" $\mathcal{A}^t$ s.t

$$\|b_{\mathcal{A}^t}\|_2 \leq \frac{1}{2} \cdot \|b_{\mathcal{A}^{t-1}}\|_2$$



Alt-Min in Theory

Convergence Rates

Linear rate of convergence

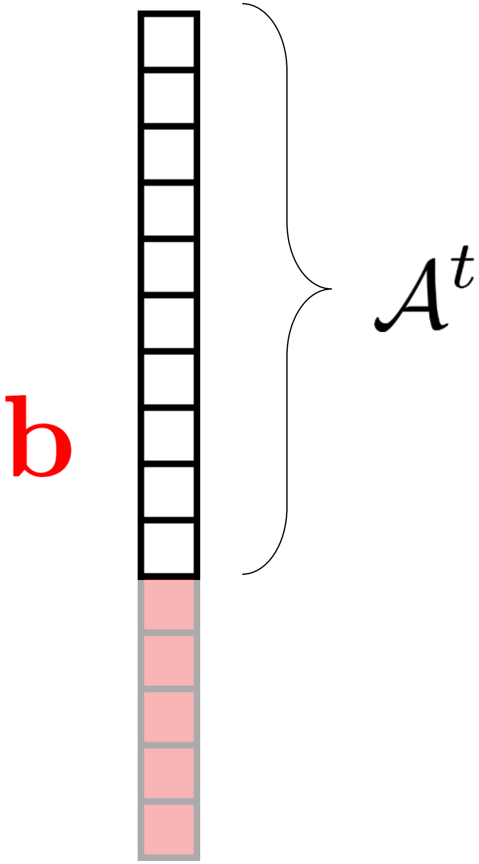
Suppose each alternation ≡ one step

After $T = \log\frac{1}{\epsilon}$ time steps

$$\left\|\mathbf{w}^T - \mathbf{w}^*\right\|_2 \leq \epsilon$$

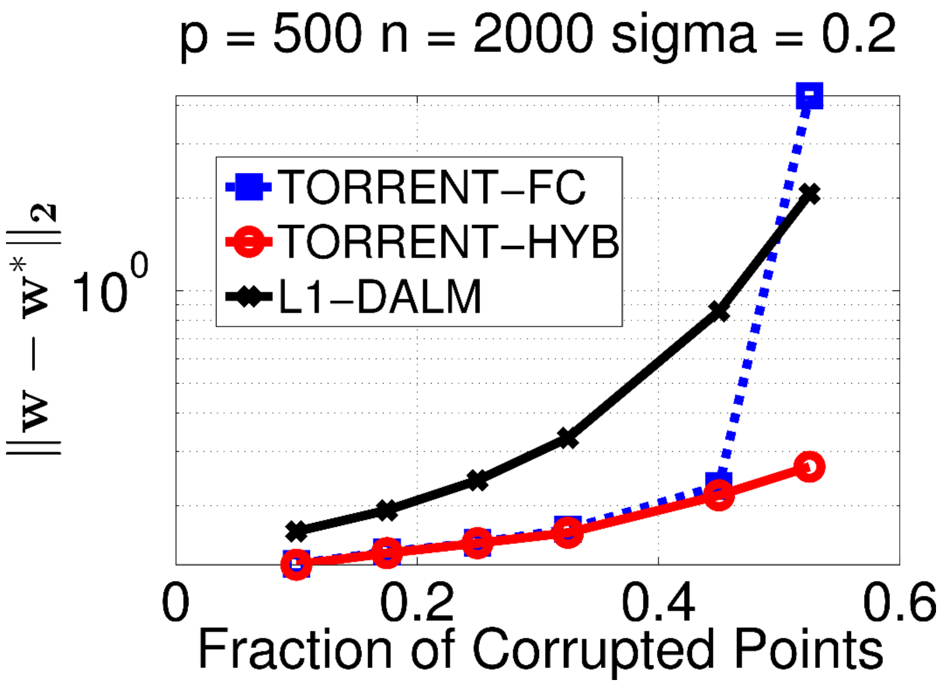
Invariant: at time t, “active set” $\mathcal{A}^t$ s.t

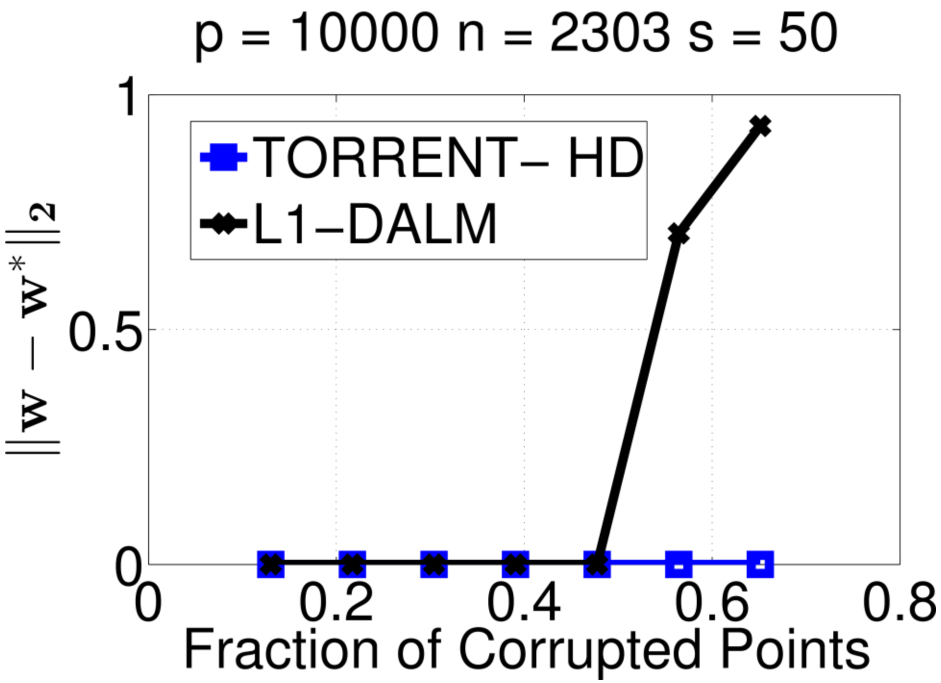
$$\|b_{\mathcal{A}^t}\|_2 \leq \frac{1}{2} \cdot \|b_{\mathcal{A}^{t-1}}\|_2$$



Alt-Min in Practice

Quality of Recovery

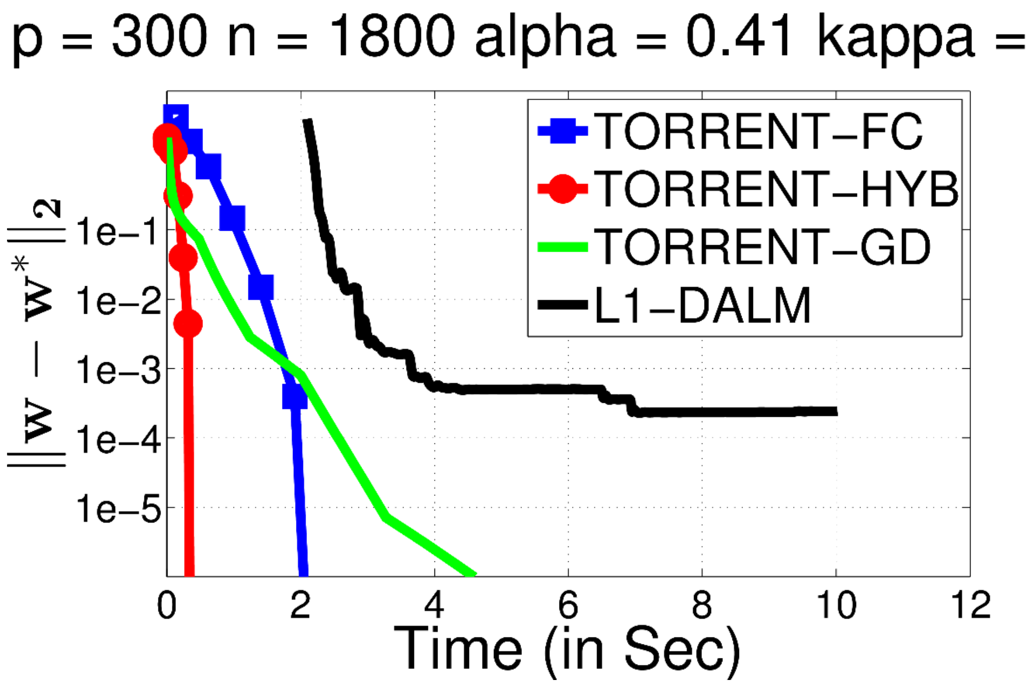


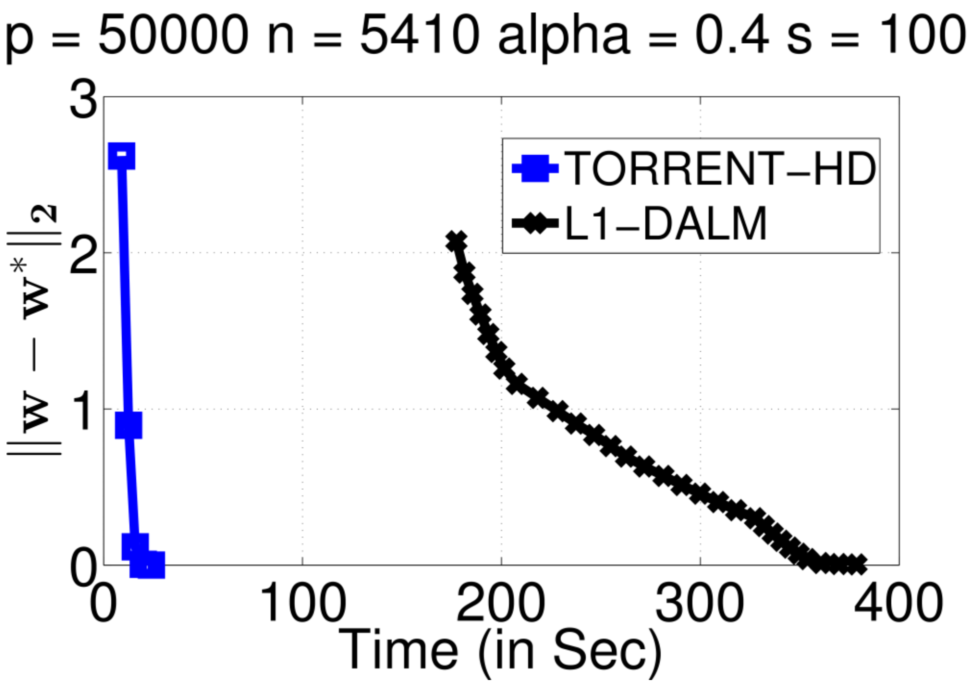


[Bhatia *et al* 2015]

Alt-Min in Practice

Speed of Recovery



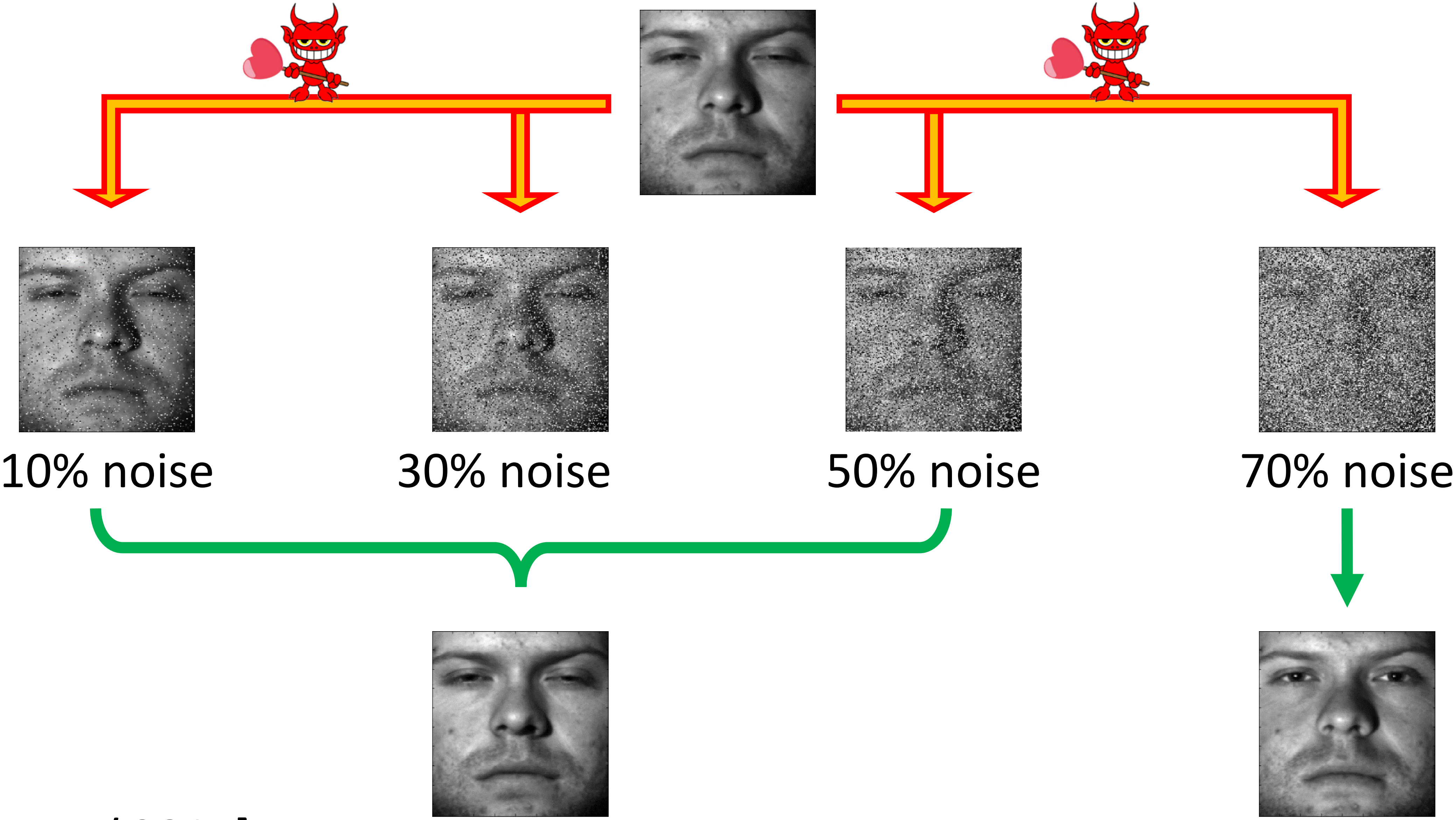


[Bhatia *et al* 2015]

Robust Regression: Application to Face Recognition

Extended Yale B dataset, 38 people, 800 images

Face Recognition



10% noise

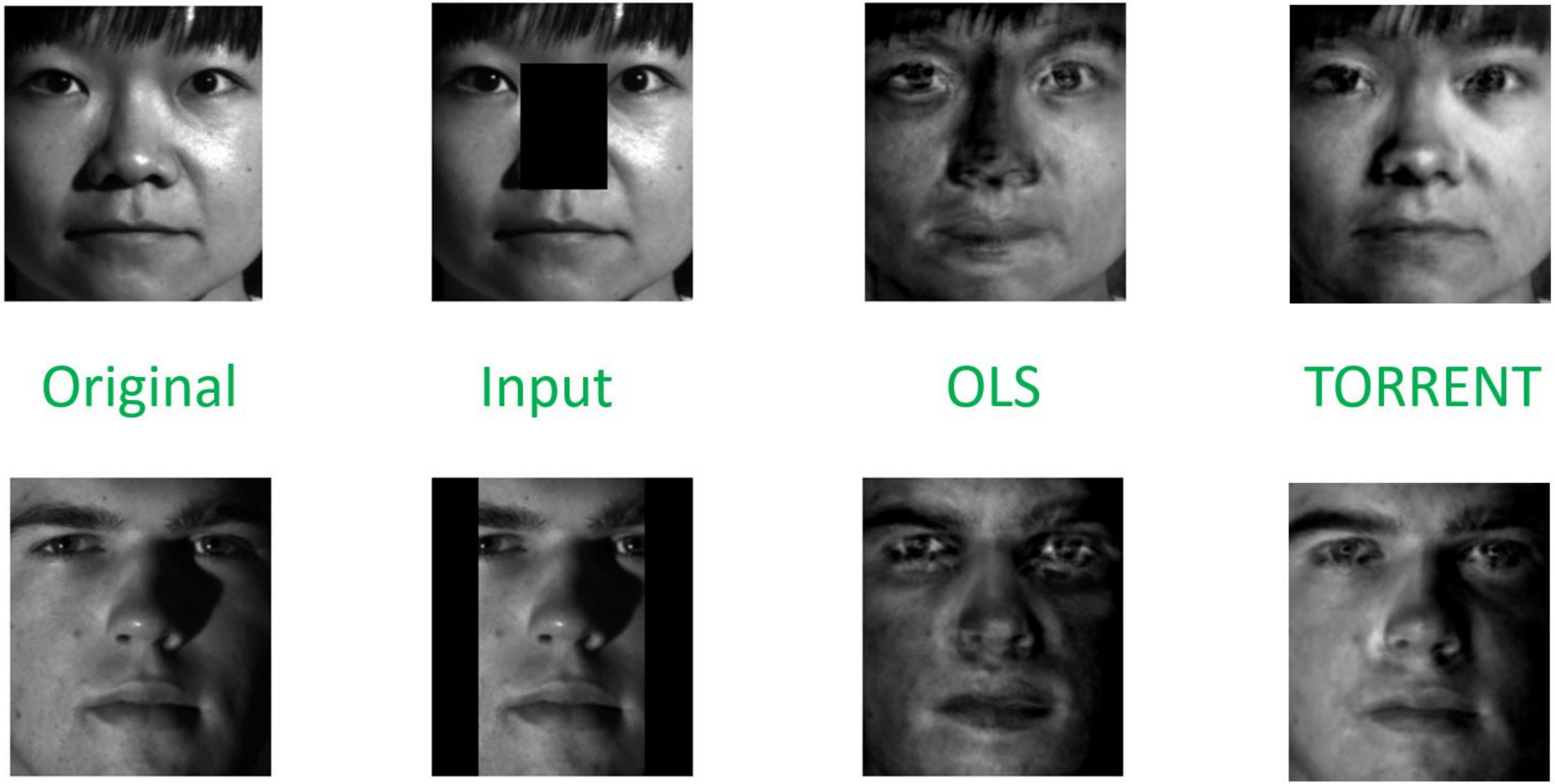
30% noise

50% noise

70% noise

[Bhatia *et al* 2015]

Image Reconstruction



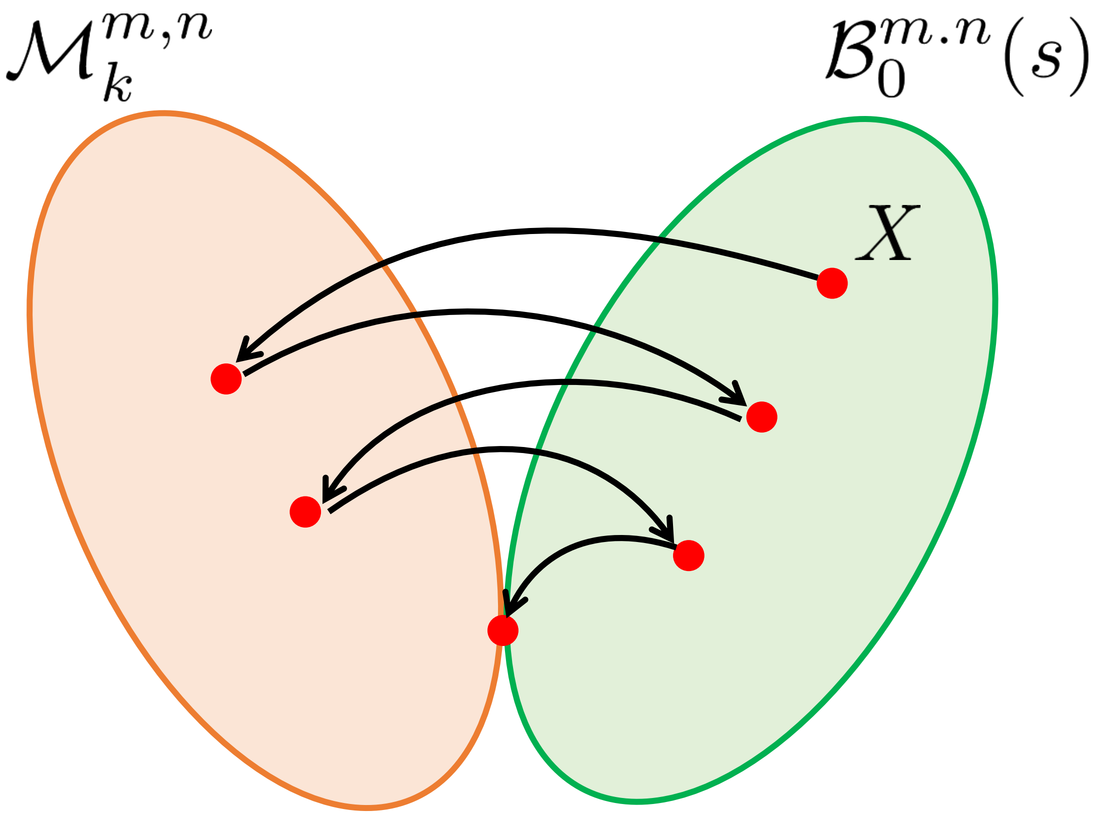
[Bhatia *et al* 2015]

Robust PCA:
A Sketch and Application to Foreground Extraction in Images

The Alternating Projection Procedure

$$\min_{\substack{L \in \mathcal{M}_k^{m,n} \\ S \in \mathcal{B}_0^{m,n}(s)}} \|X - (L + S)\|_F^2$$

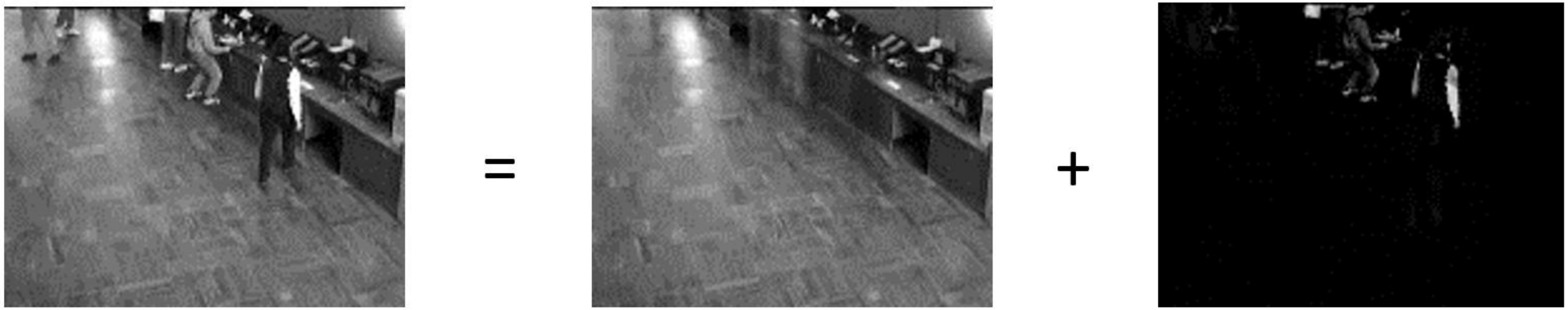
```
▷ Initialize $L^0, S^0$
▷ For $r = 1, 2, \ldots, k$
    ▷ For $t = 1, 2, \ldots, T$
        ▷ Set $s'$ appropriately
        ▷ $L^t = \Pi_{\mathcal{M}_r^{m,n}}(X - S^{t-1})$
        ▷ $S^t = \Pi_{\mathcal{B}_0^{m,n}(s')}(X - L^t)$
    ▷ $S^0 = S^T$
```



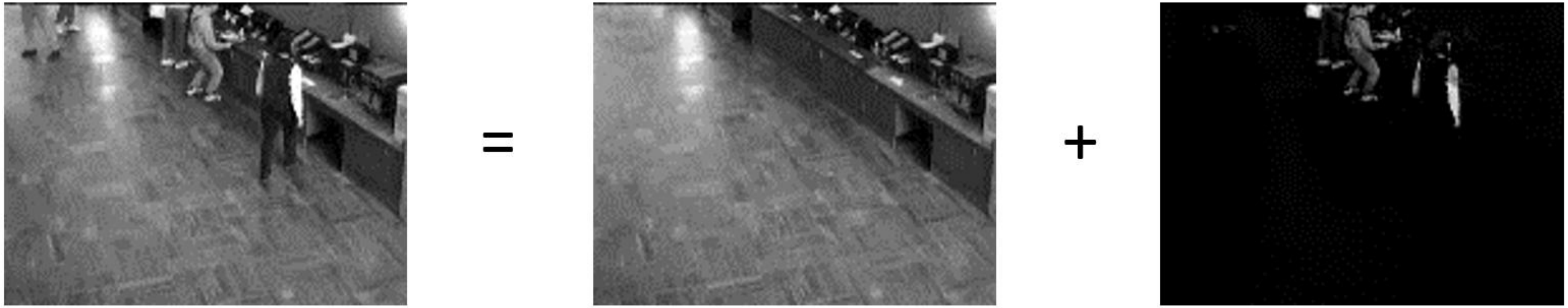
[Netrapalli *et al* 2014]

Foreground-background Separation

Convex Relaxation. Runtime: 1700 sec



Alt-Proj. Runtime: 70 sec



[Netrapalli *et al* 2014]

Concluding Comments

Non-convex optimization is an exciting area

Widespread applications

- Much better modelling of problems
- Much more scalable algorithms
- Provable guarantees

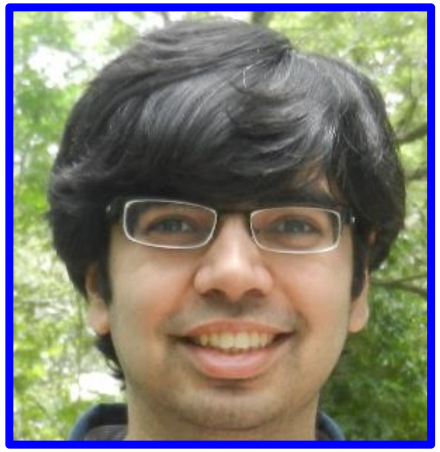
So …

- Full of opportunities
- Full of challenges

Acknowledgements

http://research.microsoft.com/en-us/projects/altmin/default.aspx

Portions of this talk were based on joint work with



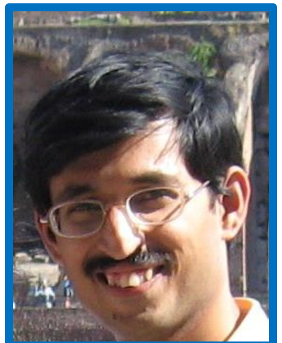
Kush Bhatia
Microsoft Research

Prateek Jain
Microsoft Research

Ambuj Tewari
U. Michigan, Ann Arbor

The Data Sciences Gang@IITK

Medha Atre



Arnab Bhattacharya

Sumit Ganguly

Purushottam Kar

Harish Karnick

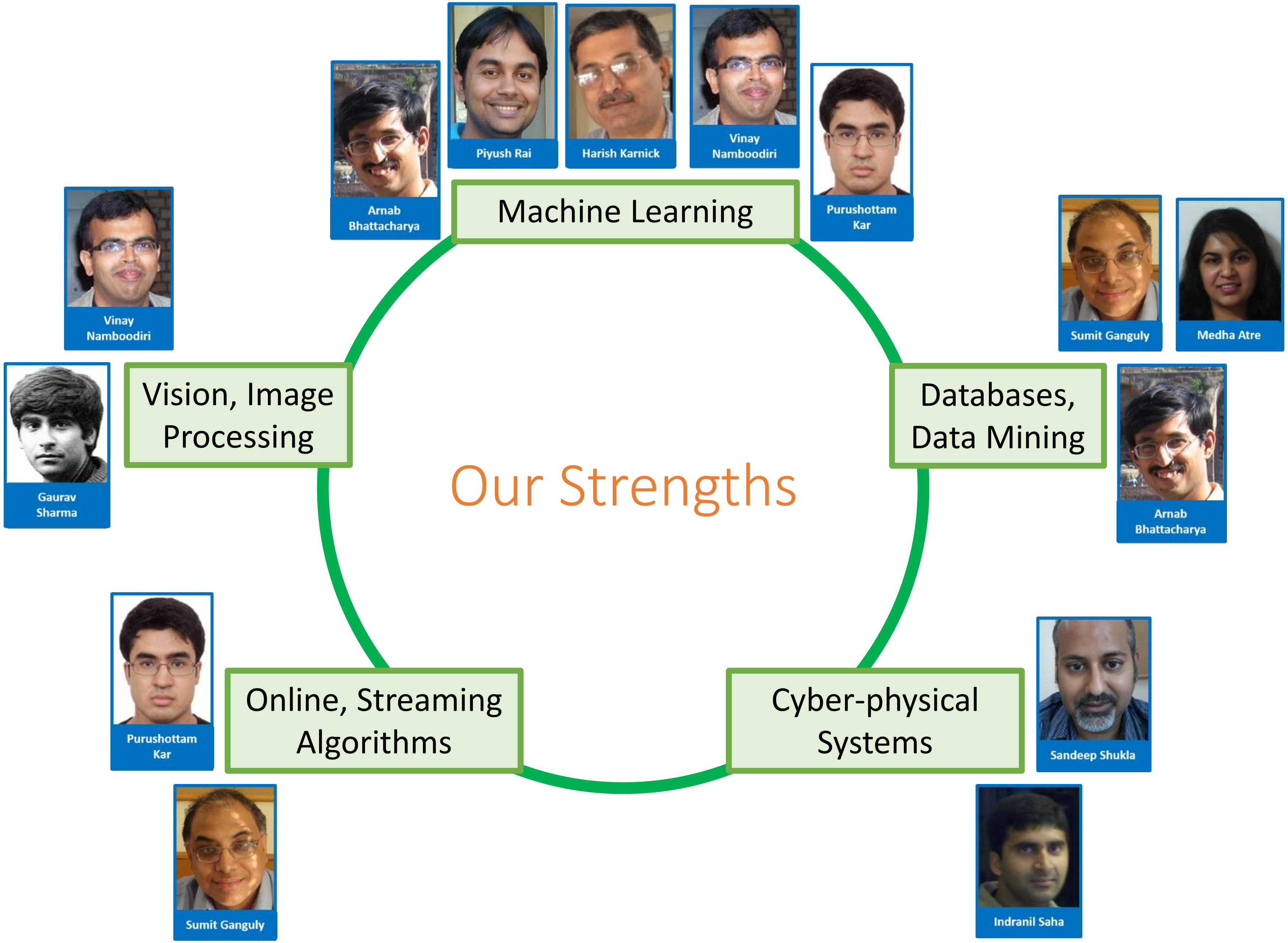
Vinay Namboodiri

Piyush Rai

Indranil Saha

Gaurav Sharma

Sandeep Shukla



Our Strengths

Machine Learning

Arnab Bhattacharya

Piyush Rai

Harish Karnick

Vinay Namboodiri

Purushottam Kar

Databases, Data Mining

Sumit Ganguly

Medha Atre

Arnab Bhattacharya

Cyber-physical Systems

Sandeep Shukla

Indranil Saha

Online, Streaming Algorithms

Purushottam Kar

Sumit Ganguly

Vision, Image Processing

Vinay Namboodiri

Gaurav Sharma

Questions?

TORRENT as an Alt-Min Procedure

- TORRENT indeed performs Alt-Min
- Two variables in TORRENT – active set $\mathcal{A}$ and model $\mathbf{w}$

$$\min_{\mathbf{w}\in\mathbb{R}^p} f(\mathcal{A},\mathbf{w}) = \sum_{i\in\mathcal{A}} (y_i - \mathbf{x}_i^\top \mathbf{w})^2$$

$$s.t. \ |\mathcal{A}| \leq n - k = (1-\alpha)\cdot n$$

- $\mathcal{A}$ encodes the complement of the corruption vector $\mathbf{b}$
- TORRENT alternates between
  - Fixing model and choosing active set
  - Fixing active set and choosing model
- Both steps reduce the residual as much as possible

Linear Regression with Corruptions

TORRENT-GD

Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i + b_i$$

Calculate $r_i = |y_i - \langle \hat{\mathbf{w}}, \mathbf{x}_i \rangle|$

Set aside k points with highest r_i

Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

$\mathcal{A}$: active points

$$\hat{\mathbf{w}} = \hat{\mathbf{w}} - \nabla \left(\sum_{i \in \mathcal{A}} (y_i - \langle \hat{\mathbf{w}}, \mathbf{x}_i \rangle)^2 \right)$$

Given remaining points, easy to "improve" $\hat{\mathbf{w}}$

Thresholding Operator-based Robust RegrEssioN meThod [Bhatia *et al*, 2015]

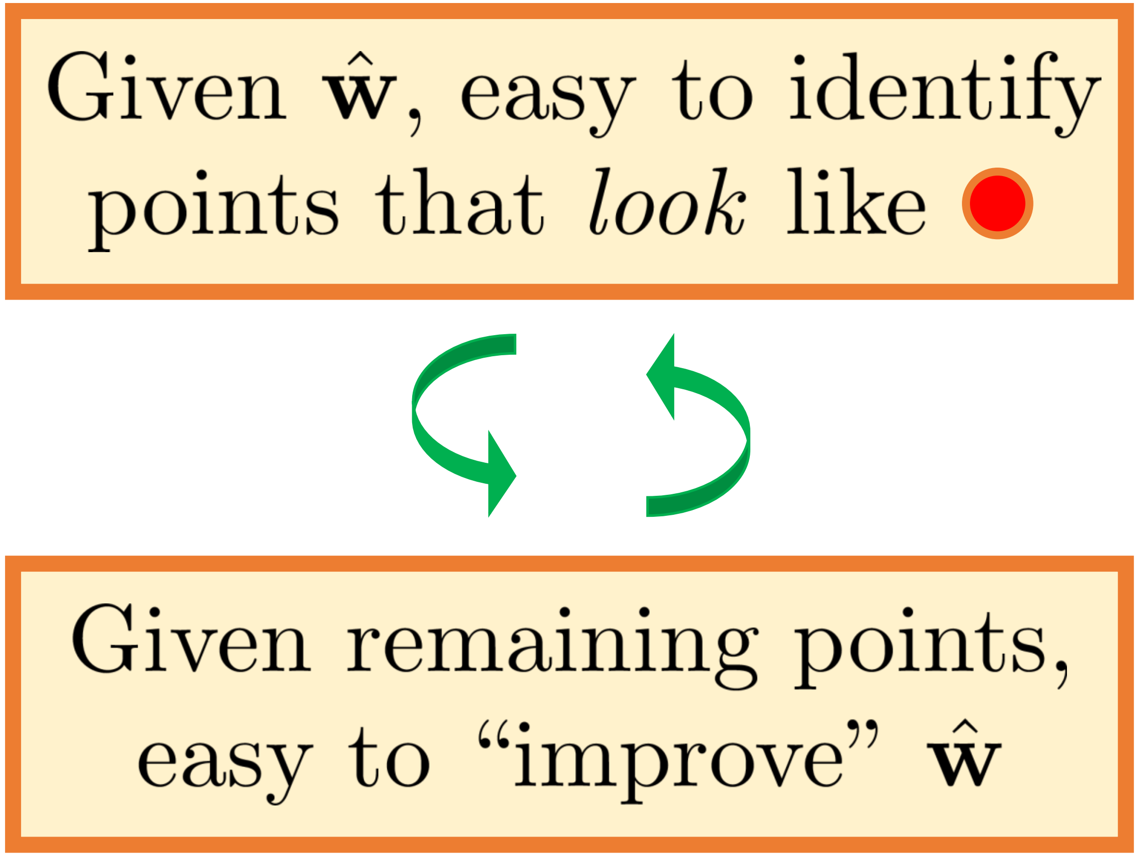
Linear Regression with Corruptions

TORRENT-HYB

Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

$$y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + e_i + b_i$$

```
If active set $\mathcal{A}$ “stable”
        execute TORRENT-FC
Else
        execute TORRENT-GD
```



Given $\hat{\mathbf{w}}$, easy to identify points that *look* like ●

Given remaining points, easy to “improve” $\hat{\mathbf{w}}$

Thresholding Operator-based Robust RegrEssioN meThod [Bhatia *et al*, 2015]