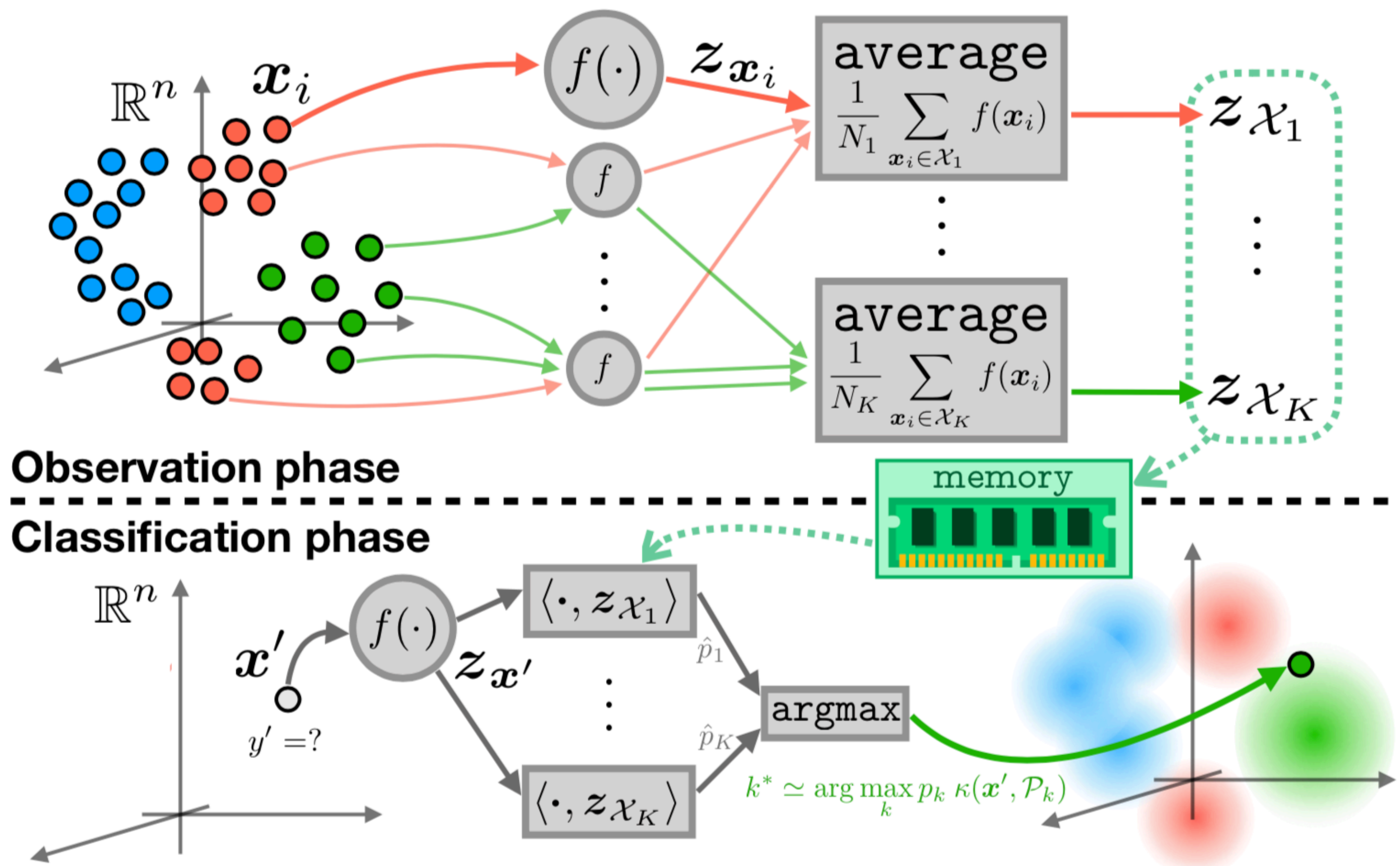
Compressive Classification

(Machine Learning without learning)



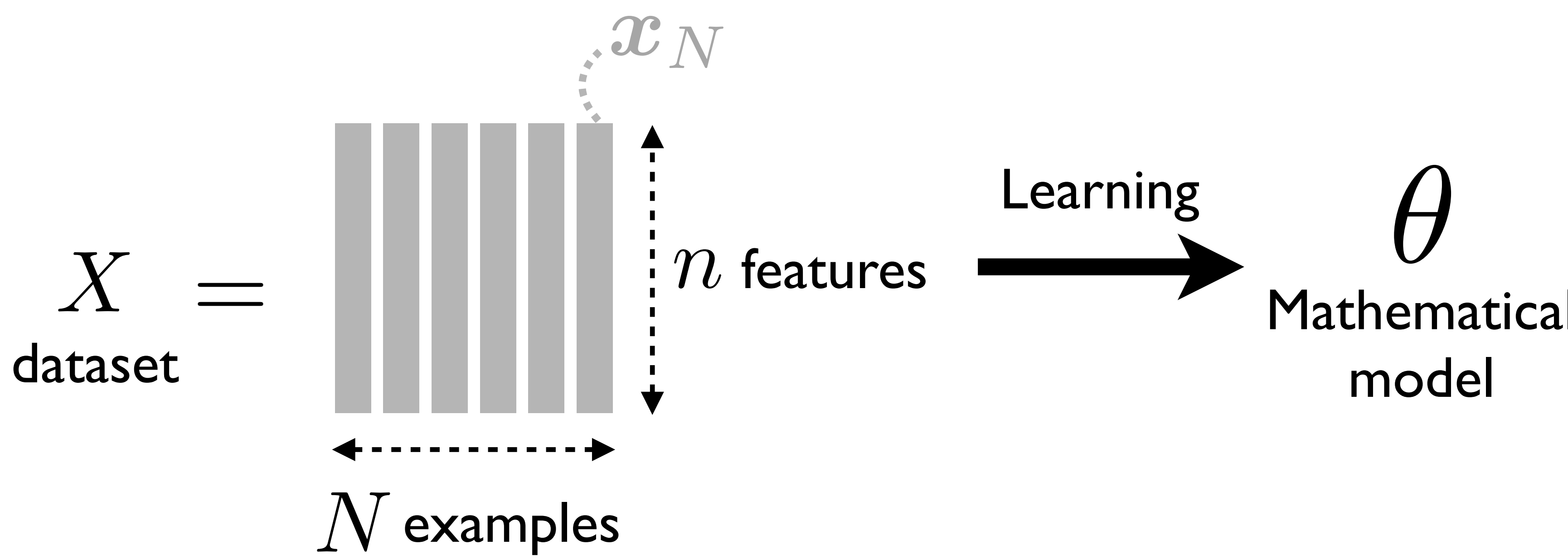
Vincent Schellekens Laurent Jacques

UCLouvain

ISP GROUP Image & Signal Processing

fnrs LA LIBERTÉ DE CHERCHER

Machine Learning

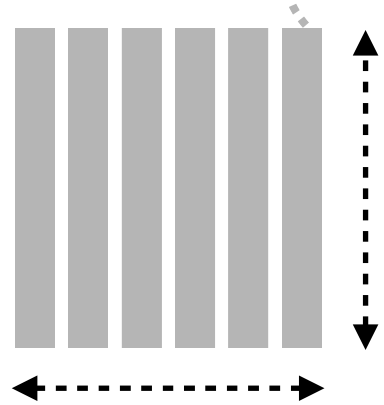


Machine Learning

Unsupervised

X dataset $=$

x_N



n features

N examples

Learning →

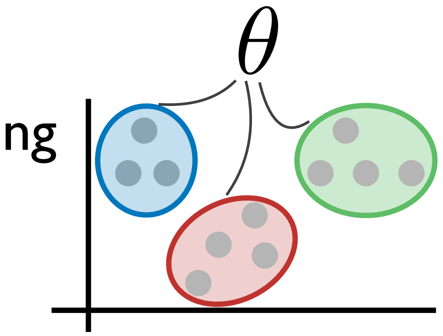
θ

Mathematical model

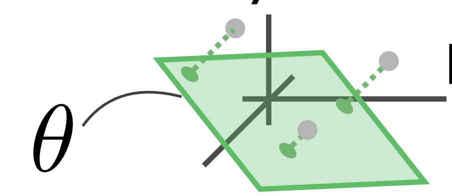
"explain" the data

E.g.,

- Clustering

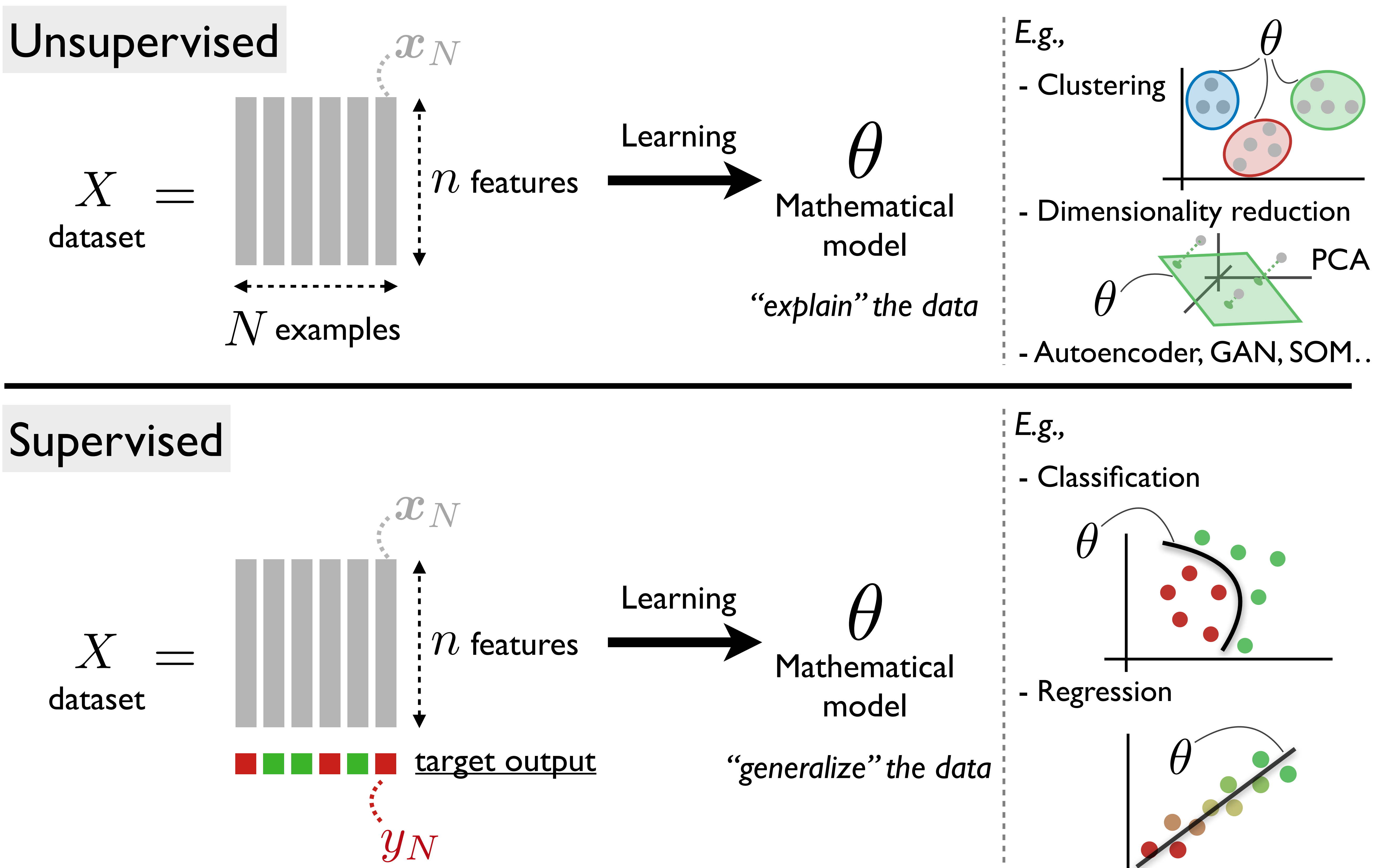


- Dimensionality reduction



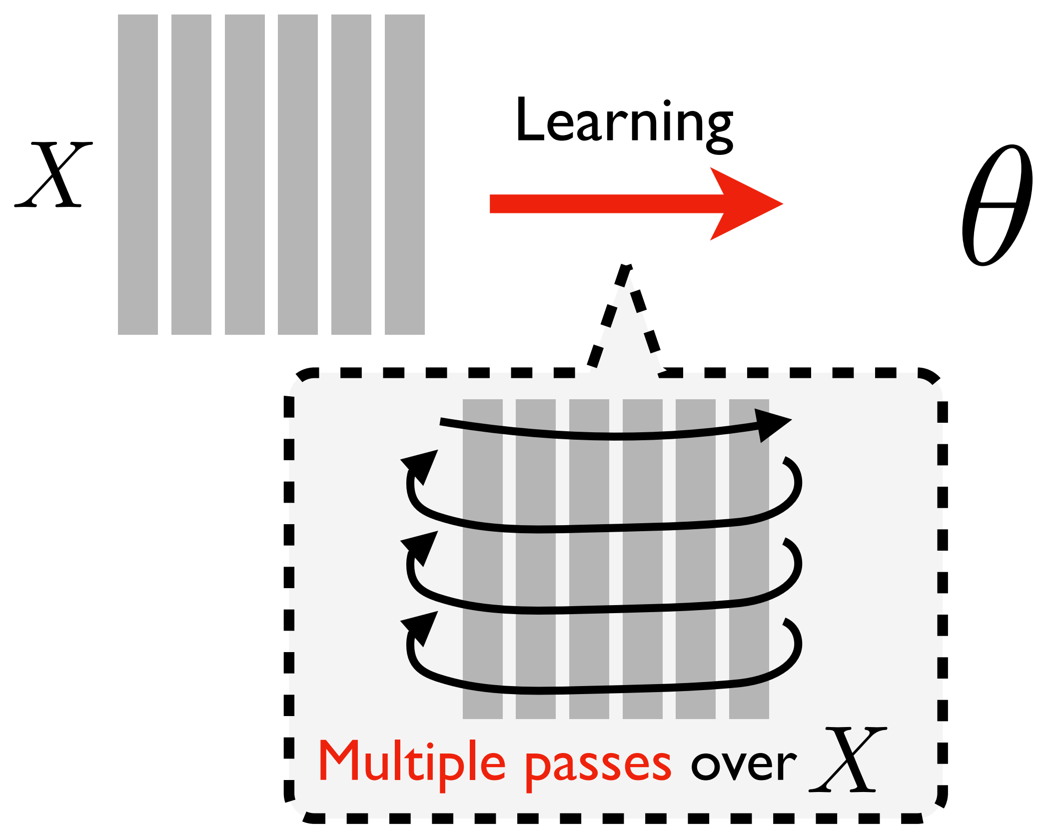
- Autoencoder, GAN, SOM…

Machine Learning



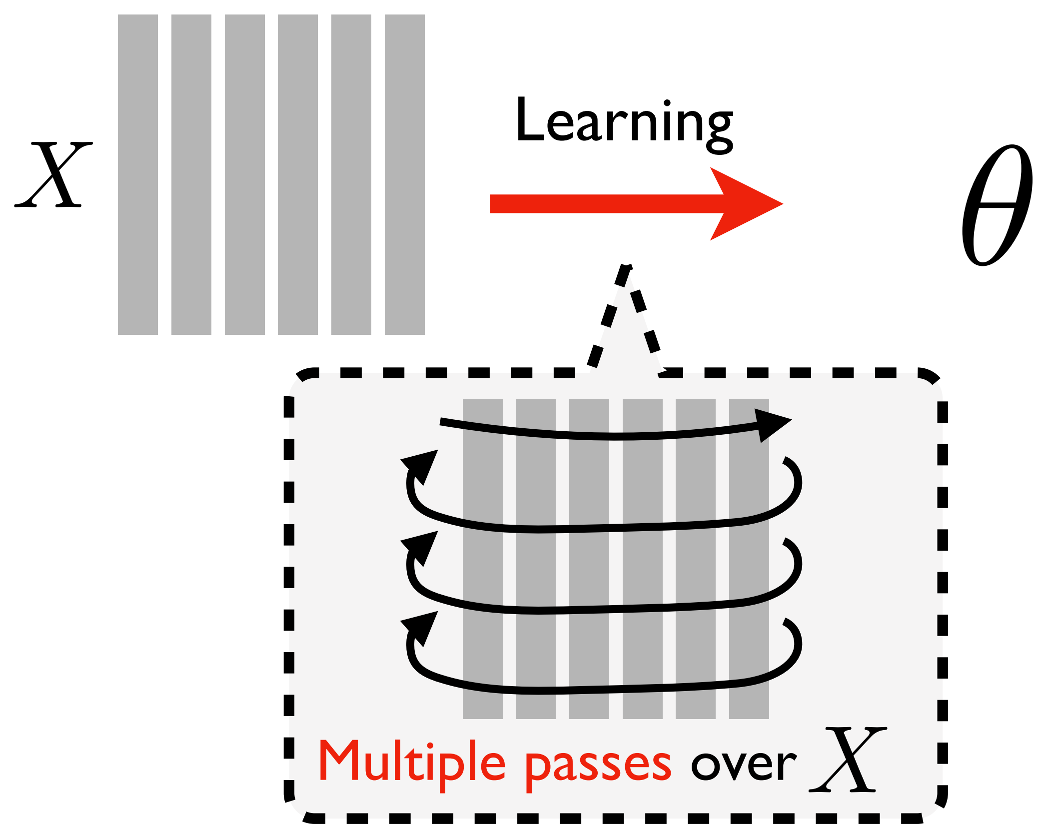
Compressive Learning

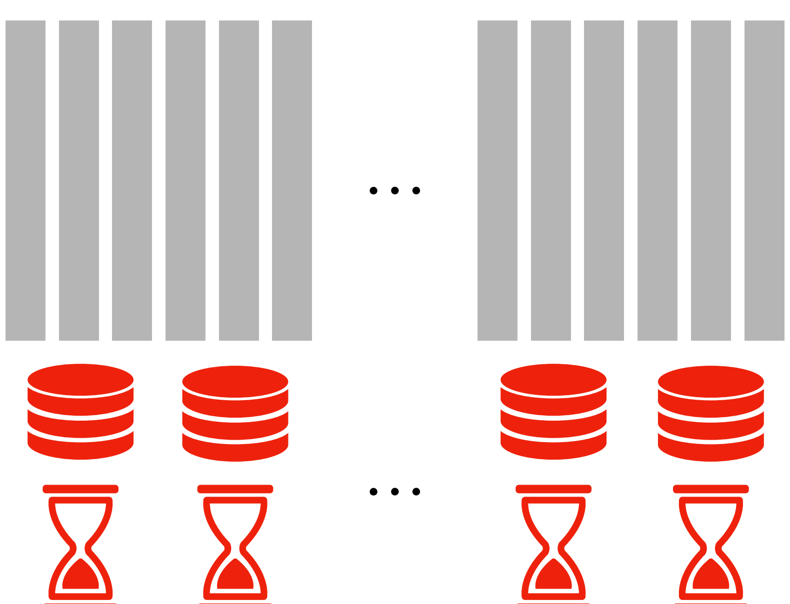
Usual machine learning



Compressive Learning

Usual machine learning



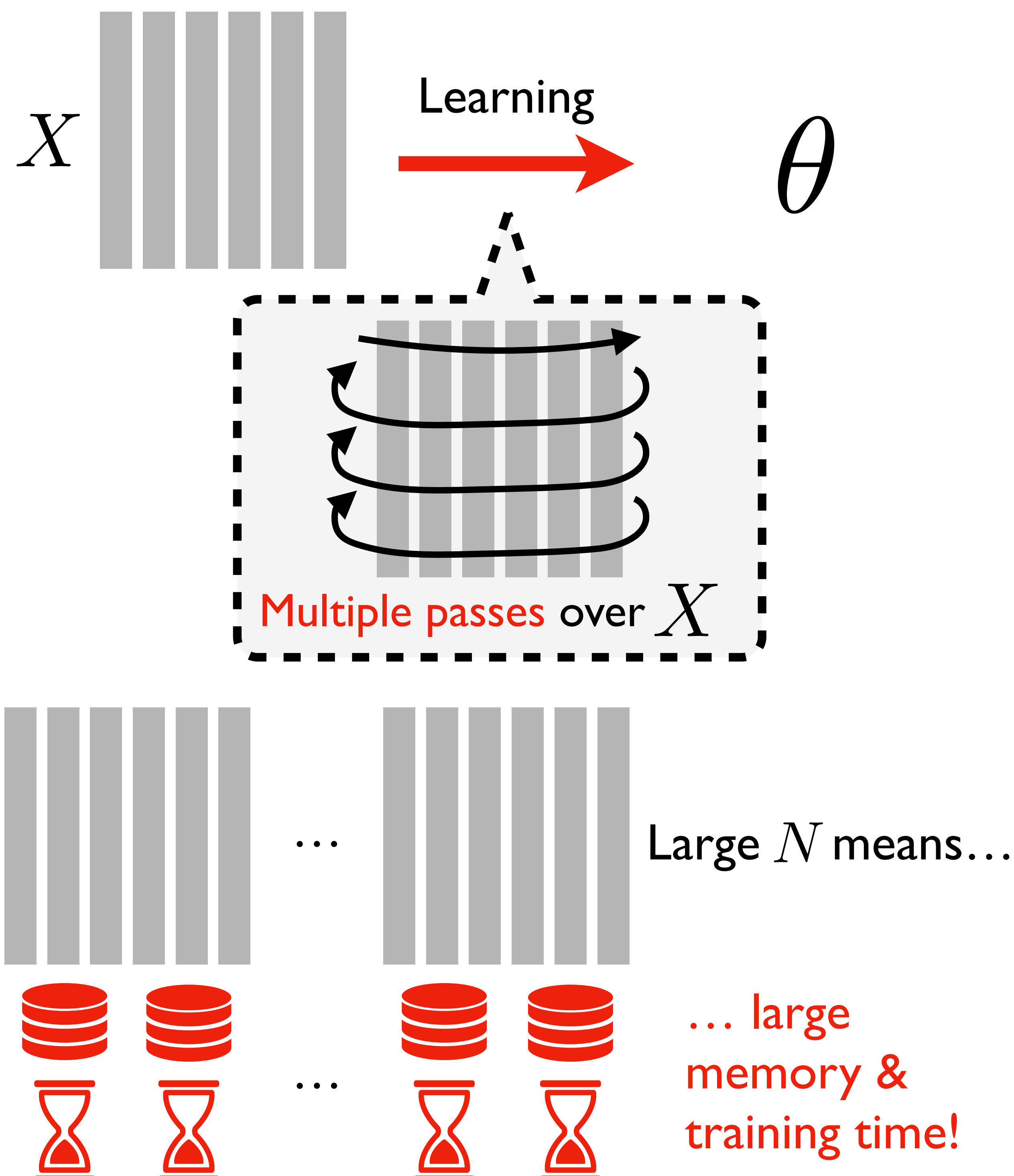


Large N means…

… large memory & training time!

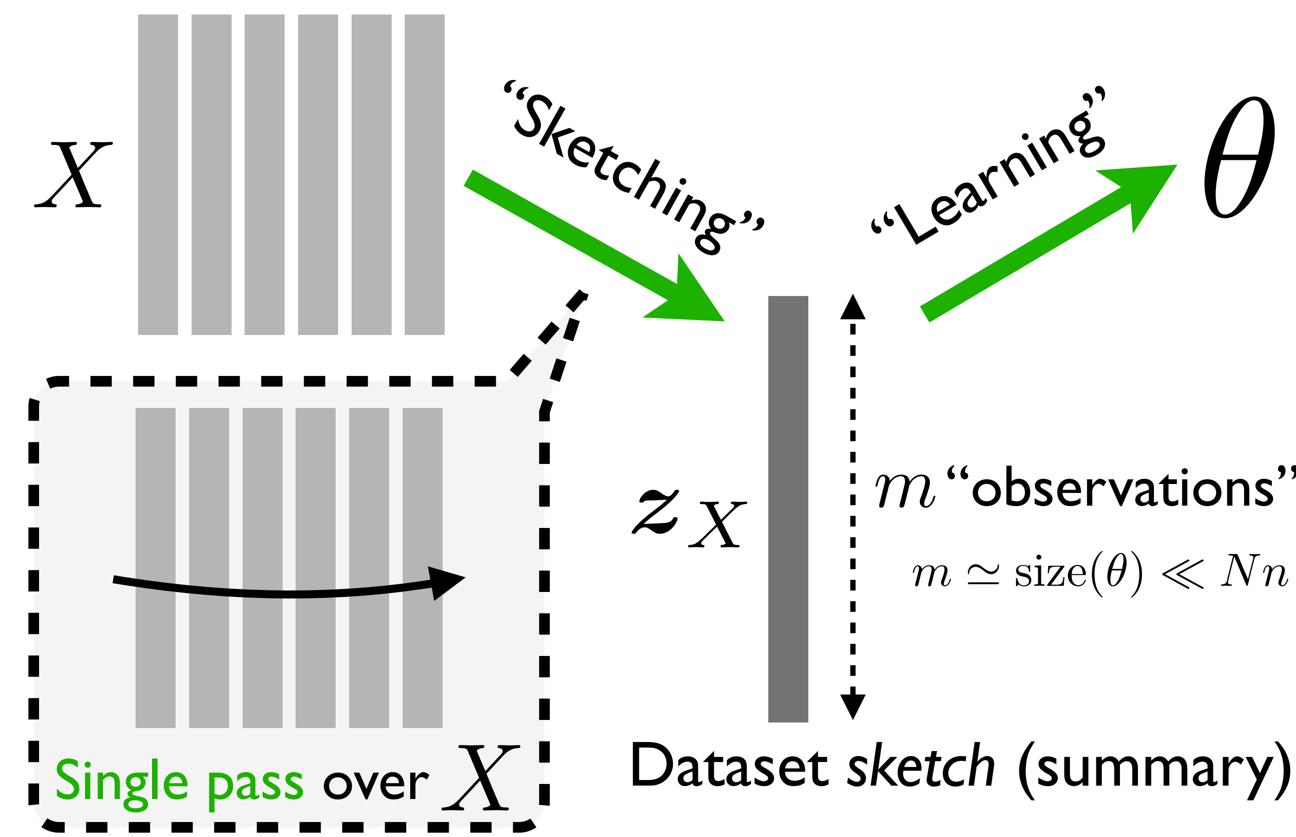
Compressive Learning

Usual machine learning



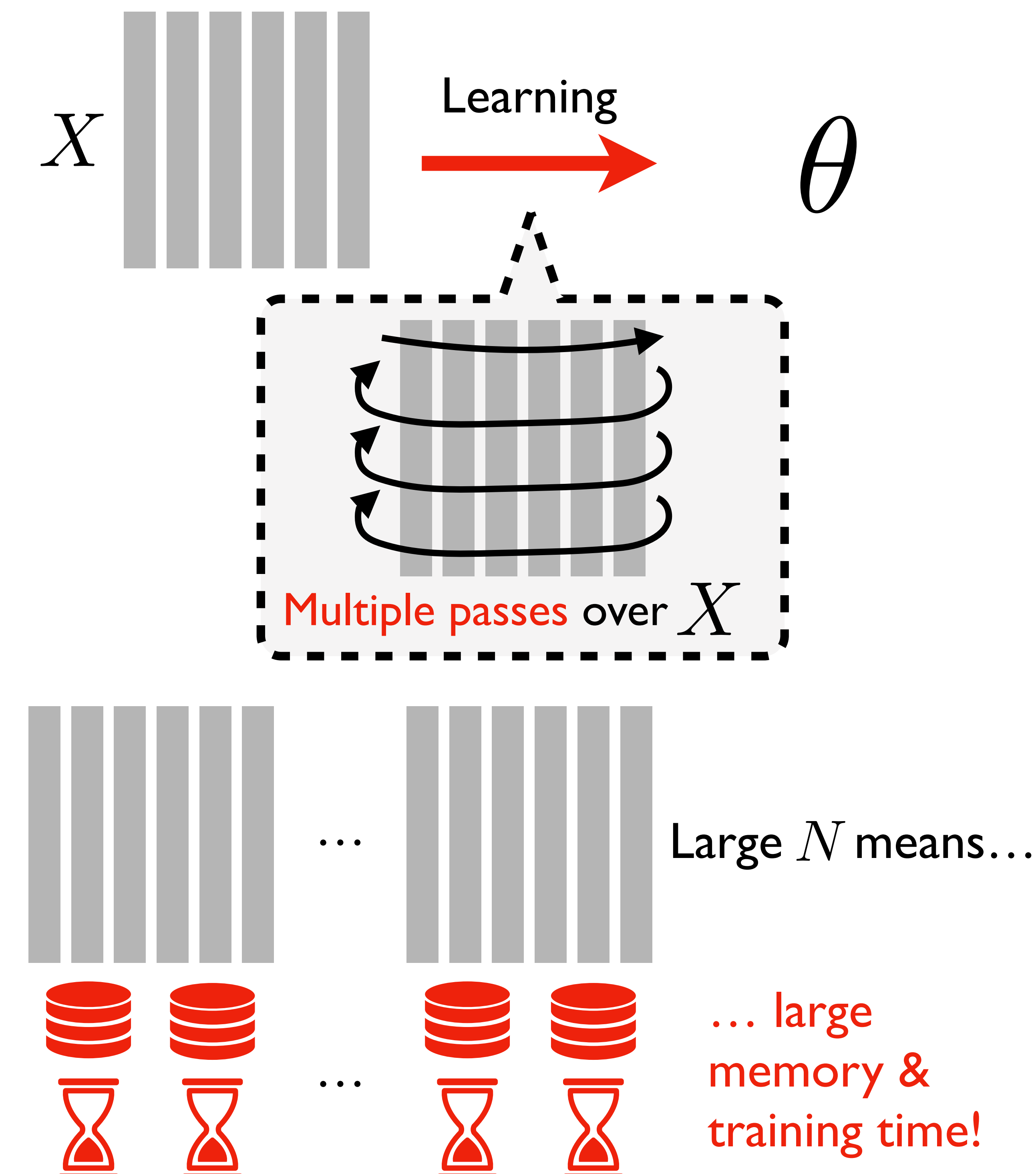
Compressive Learning

e.g., [Gribonval-CL]



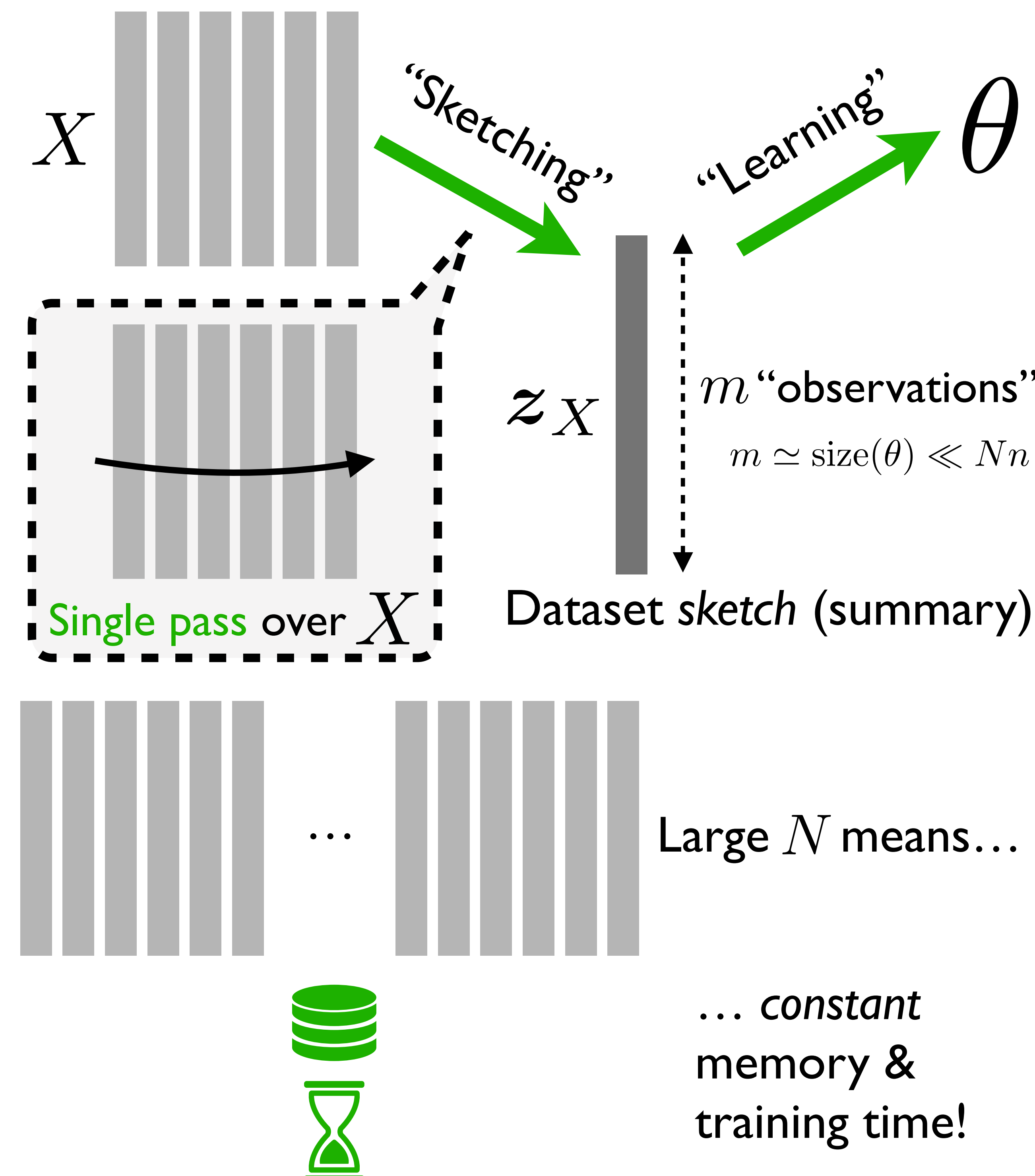
Compressive Learning

Usual machine learning



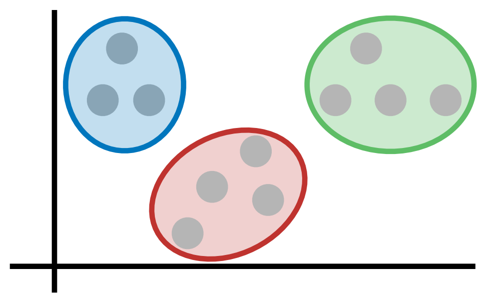
Compressive Learning

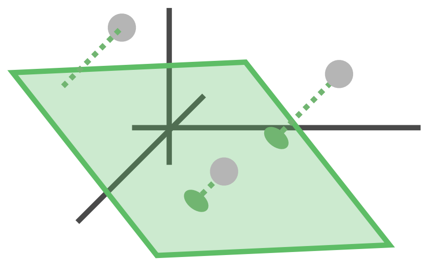
e.g., [Gribonval-CL]



Previously on Compressive Learning…

Unsupervised ML

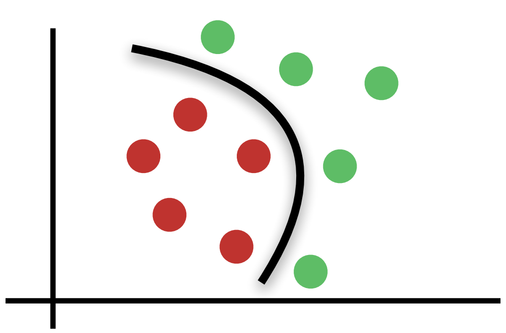


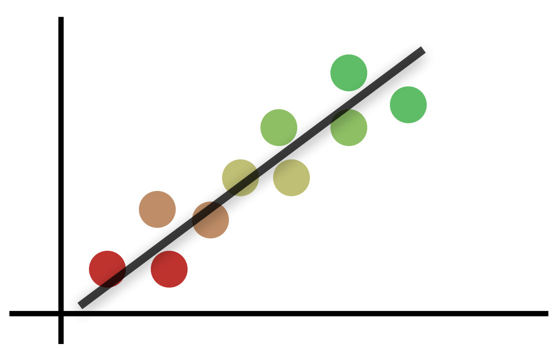


Unsupervised Compressive Learning

- Compressive K-Means [Keriven-CKM]
- Compressive GMM estimation [Keriven-GMM]
- Compressive PCA [Gribonval-CL]

Supervised ML



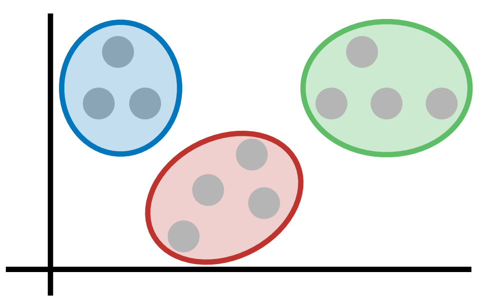


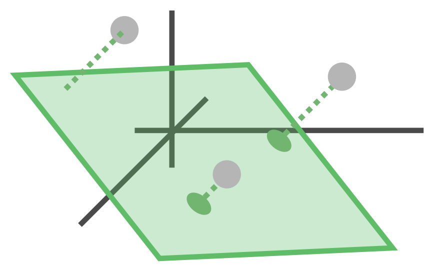
Supervised Compressive Learning

????

In this talk…

Unsupervised ML

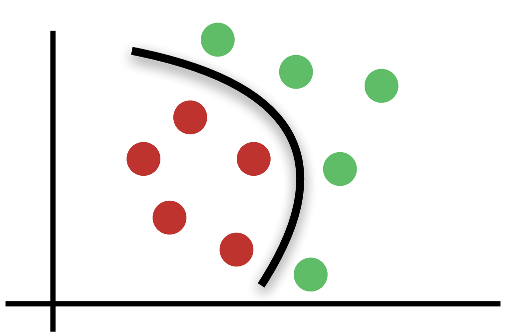


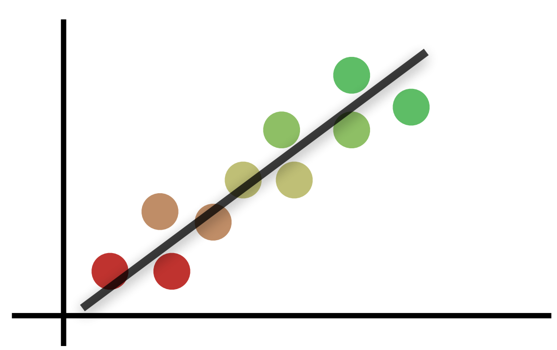


Unsupervised Compressive Learning

- Compressive K-Means [Keriven-CKM]
- Compressive GMM estimation [Keriven-GMM]
- Compressive PCA [Gribonval-CL]

Supervised ML





Supervised Compressive Learning

Compressive Classification

(a proof of concept)

In this talk…

Unsupervised ML

Unsupervised Compressive Learning

- Compressive K-Means [Keriven-CKM]
- Compressive GMM estimation [Keriven-GMM]
- Compressive PCA [Gribonval-CL]

Preliminary 1:

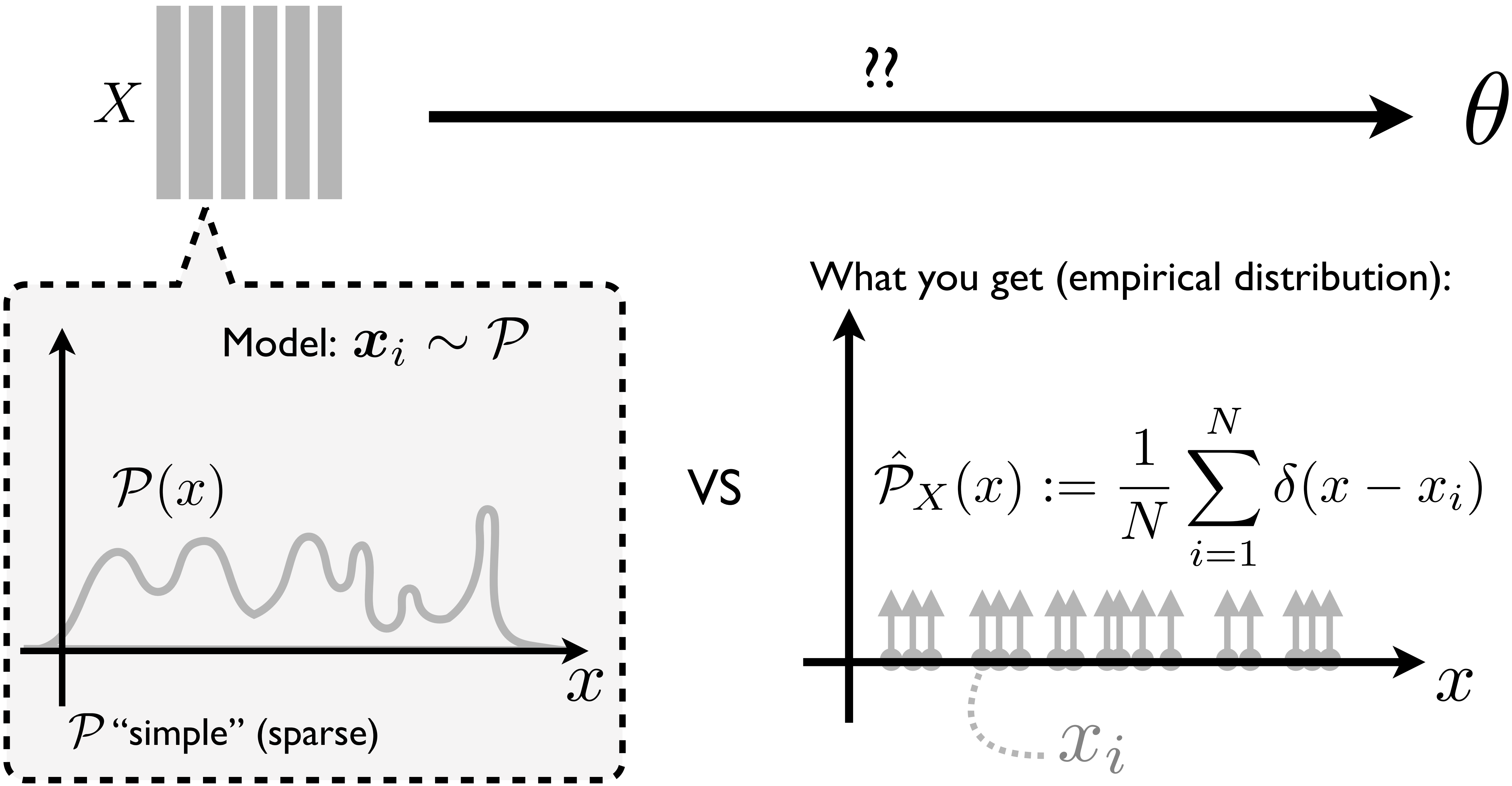
(Unsupervised) Compressive Learning Basics

Compressive Learning

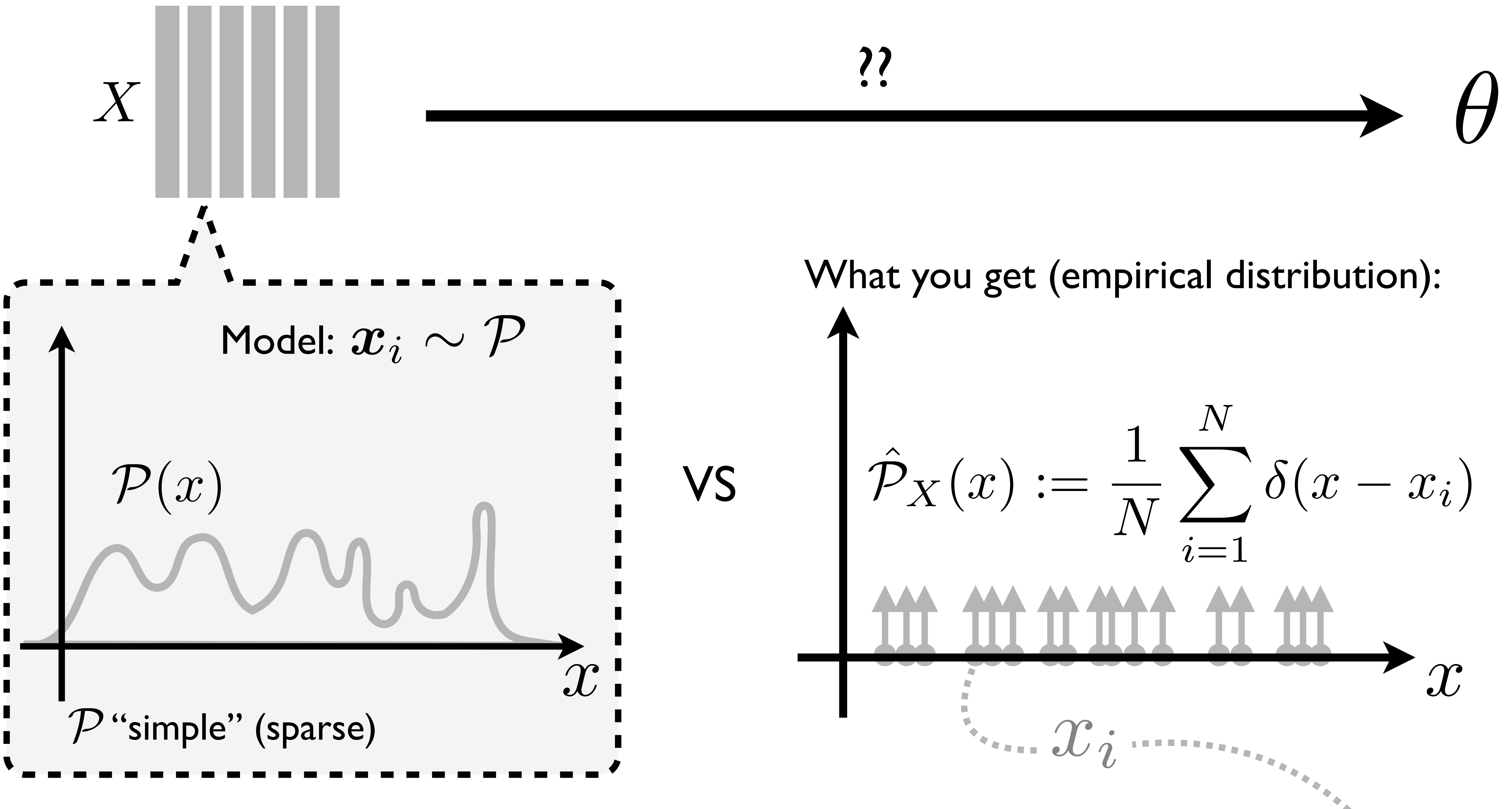
Compressive Classification

(a proof of concept)

Compressing a dataset



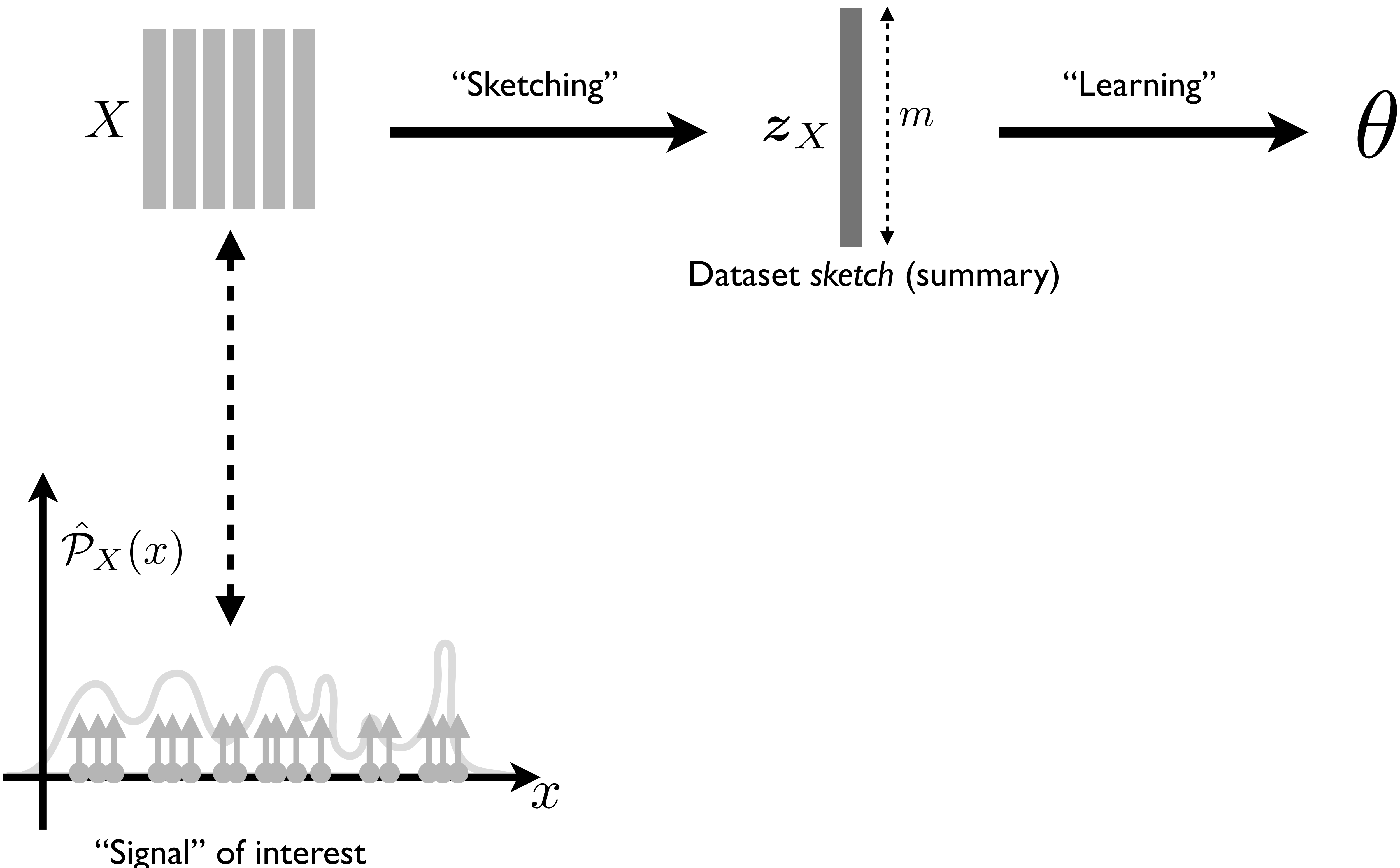
Compressing a ~~dataset~~ distribution



$$\hat{\mathcal{P}}_X(x) := \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i)$$

We don't care about individual realisations
Object of interest: the distribution $\mathcal{P}$!

Compressing a dataset distribution



Compressing a dataset

distribution

CS-inspired!

X $\xrightarrow{\text{“Sketching”}}$ $\boldsymbol{z}_X$ (m) $\xrightarrow{\text{“Learning”}}$ θ

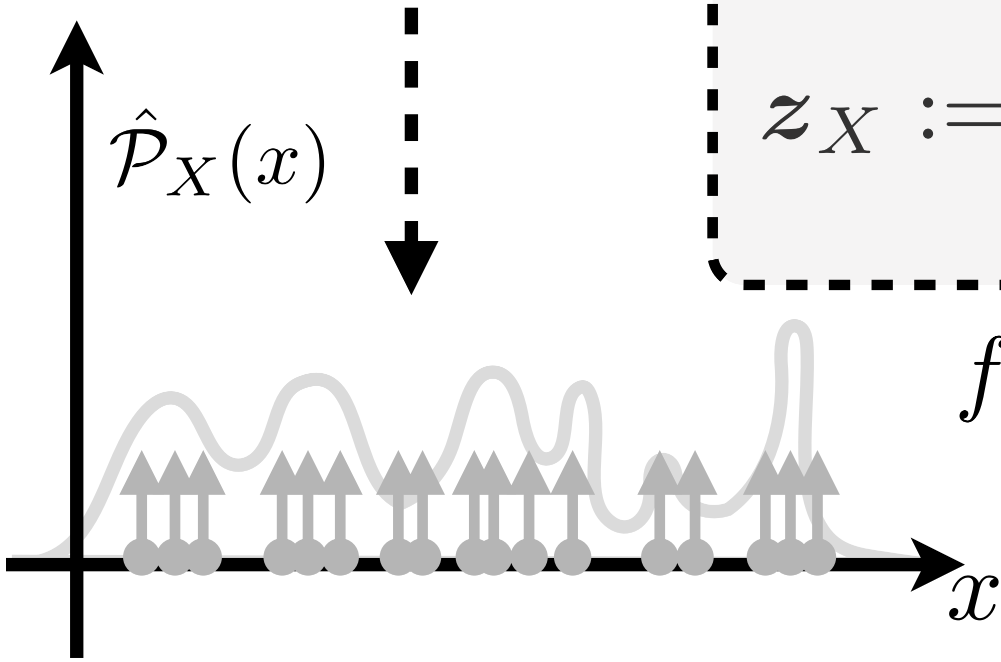
Generalized moments
Linear sensing

Moment matching
Inverse problem

Dataset *sketch* (summary)

$$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} f(\boldsymbol{x})$$
$$\simeq$$
$$\boldsymbol{z}_X := \mathcal{A}(\hat{\mathcal{P}}_X) = \frac{1}{N}\sum_{\boldsymbol{x}_i \in X} f(\boldsymbol{x}_i) \in \mathbb{C}^m$$

f computes m “features” of the data, to be averaged (“pooled”)



“Signal” of interest

Compressing a ~~dataset~~ distribution

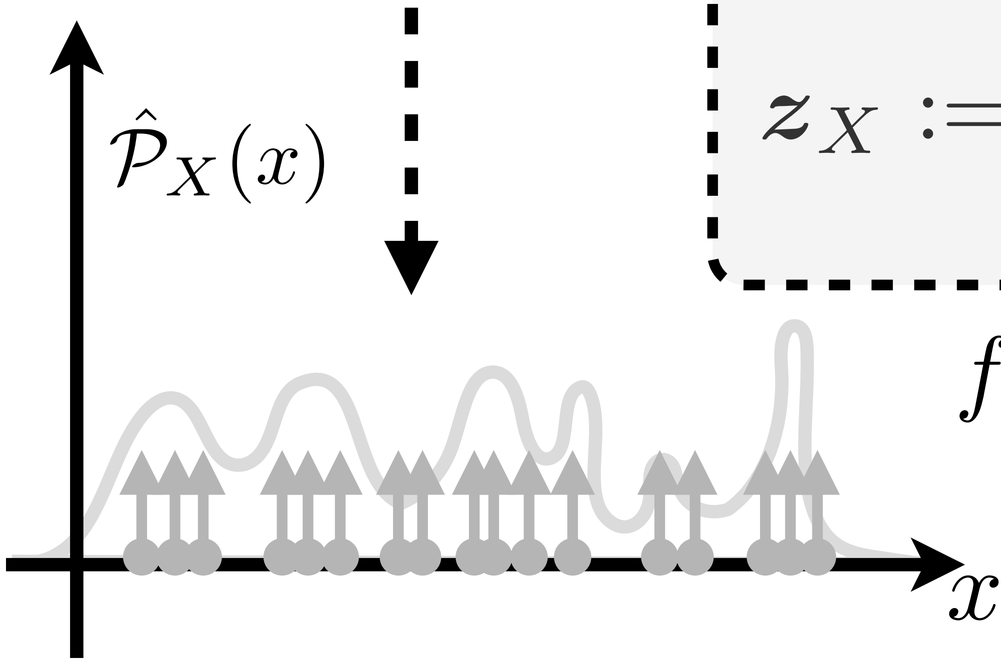
CS-inspired!

X → "Sketching" (Generalized moments; Linear sensing) → $\boldsymbol{z}_X$ (m) → "Learning" (Moment matching; Inverse problem) → θ

Dataset *sketch* (summary)

$$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} f(\boldsymbol{x})$$
$$\simeq$$
$$\boldsymbol{z}_X := \mathcal{A}(\hat{\mathcal{P}}_X) = \frac{1}{N}\sum_{\boldsymbol{x}_i \in X} f(\boldsymbol{x}_i) \in \mathbb{C}^m$$

f computes m "features" of the data, to be averaged ("pooled")



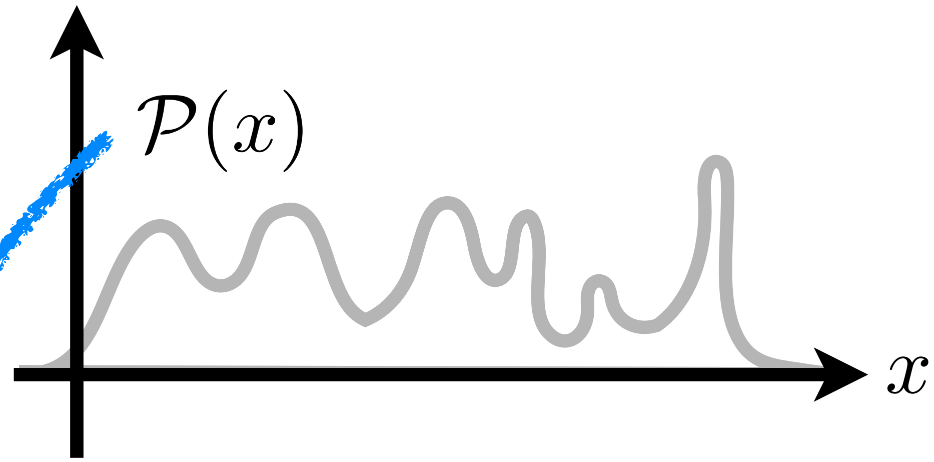
"Signal" of interest

E.g. Random Fourier Features

$$f(\boldsymbol{x})_{\mathtt{RFF}} = \left[\exp(\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x})\right]_{j=1}^m$$

Geometric interpretation

[Smola]



"Mean Map"

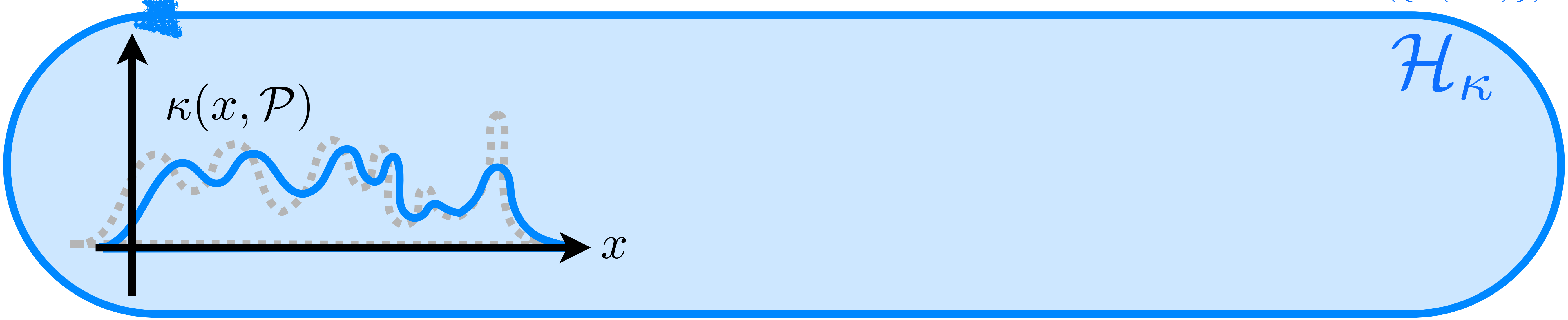
$\kappa(\cdot, \mathcal{P}) := \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{P}} \kappa(\cdot, \boldsymbol{x}')$

f approximates a kernel

$$\langle f(\boldsymbol{x}), f(\boldsymbol{x}') \rangle \simeq \kappa(\boldsymbol{x}, \boldsymbol{x}')$$

associated with a RKHS $\mathcal{H}_\kappa$

"$\mathcal{H}_\kappa = \operatorname{span}\left(\{\kappa(\cdot, \boldsymbol{u})\}\right)$"



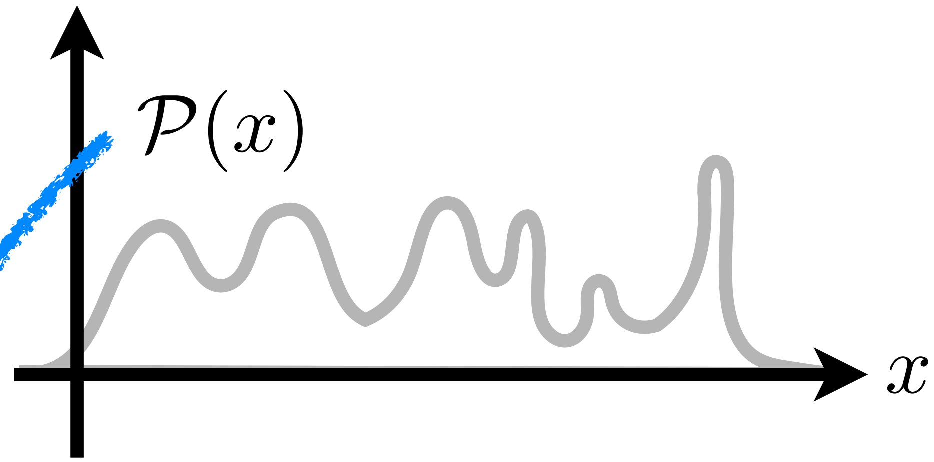
Geometric interpretation

$$f(\boldsymbol{x})_{\text{RFF}} = \left[\exp(\mathrm{i}\boldsymbol{\omega}_j^T\boldsymbol{x})\right]_{j=1}^{m} \quad \text{with } \boldsymbol{\omega}_j \sim \Lambda:$$

$$\langle f(\boldsymbol{x})_{\text{RFF}}, f(\boldsymbol{x}')_{\text{RFF}} \rangle$$

$$\simeq$$

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{x}') = (F\Lambda)(\boldsymbol{u})$$



"Mean Map"

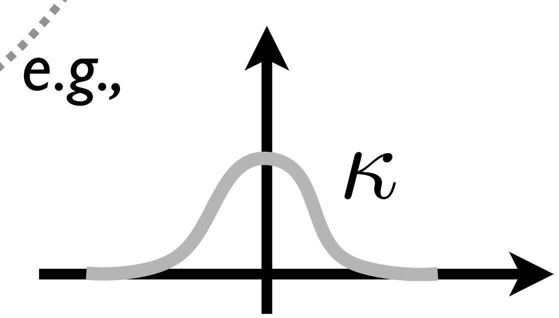
$\kappa(\cdot, \mathcal{P}) := \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{P}} \kappa(\cdot, \boldsymbol{x}')$

f approximates a kernel

$$\langle f(\boldsymbol{x}), f(\boldsymbol{x}') \rangle \simeq \kappa(\boldsymbol{x}, \boldsymbol{x}')$$

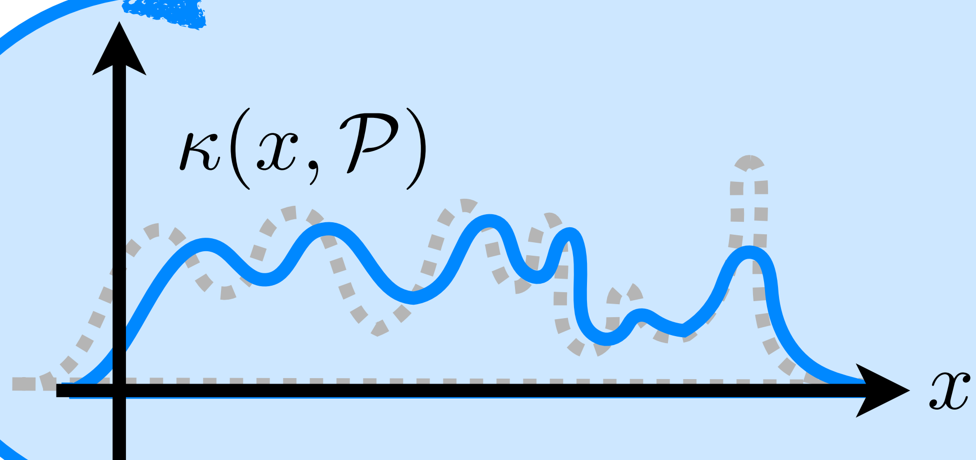
associated with a RKHS $\mathcal{H}_\kappa$

e.g.,

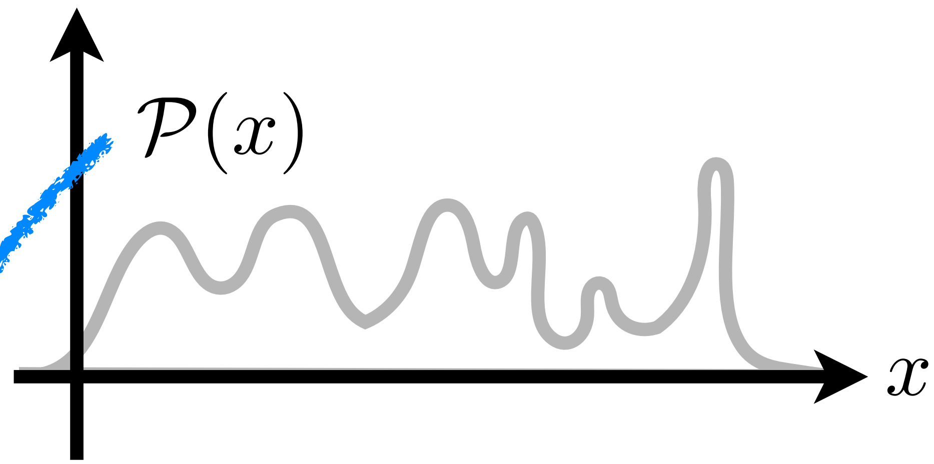


"$\mathcal{H}_\kappa = \mathrm{span}\left(\{\kappa(\cdot, \boldsymbol{u})\}\right)$"

$\mathcal{H}_\kappa$



Geometric interpretation



"Mean Map"

$\kappa(\cdot, \mathcal{P}) := \mathbb{E}_{\boldsymbol{x}' \sim \mathcal{P}} \kappa(\cdot, \boldsymbol{x}')$

f approximates a kernel

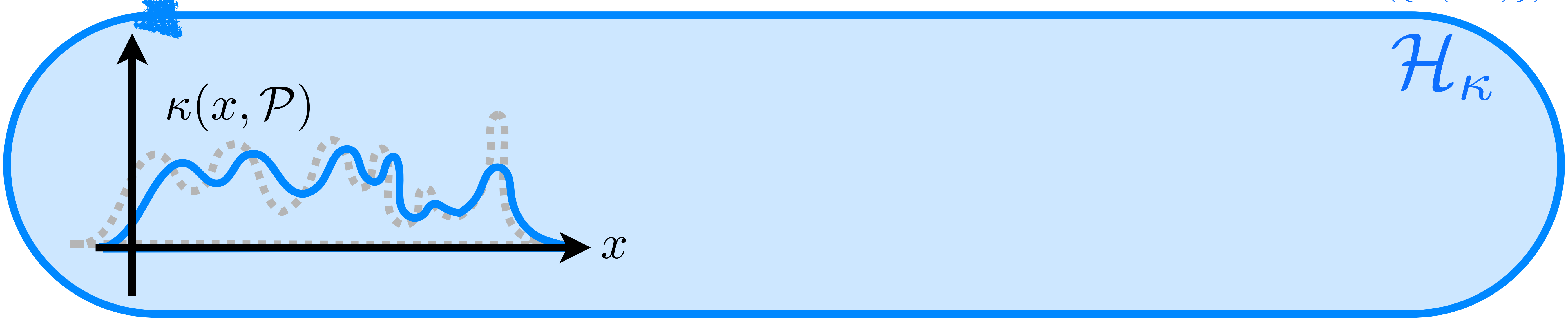
$$\langle f(\boldsymbol{x}), f(\boldsymbol{x}') \rangle \simeq \kappa(\boldsymbol{x}, \boldsymbol{x}')$$

associated with a RKHS $\mathcal{H}_\kappa$

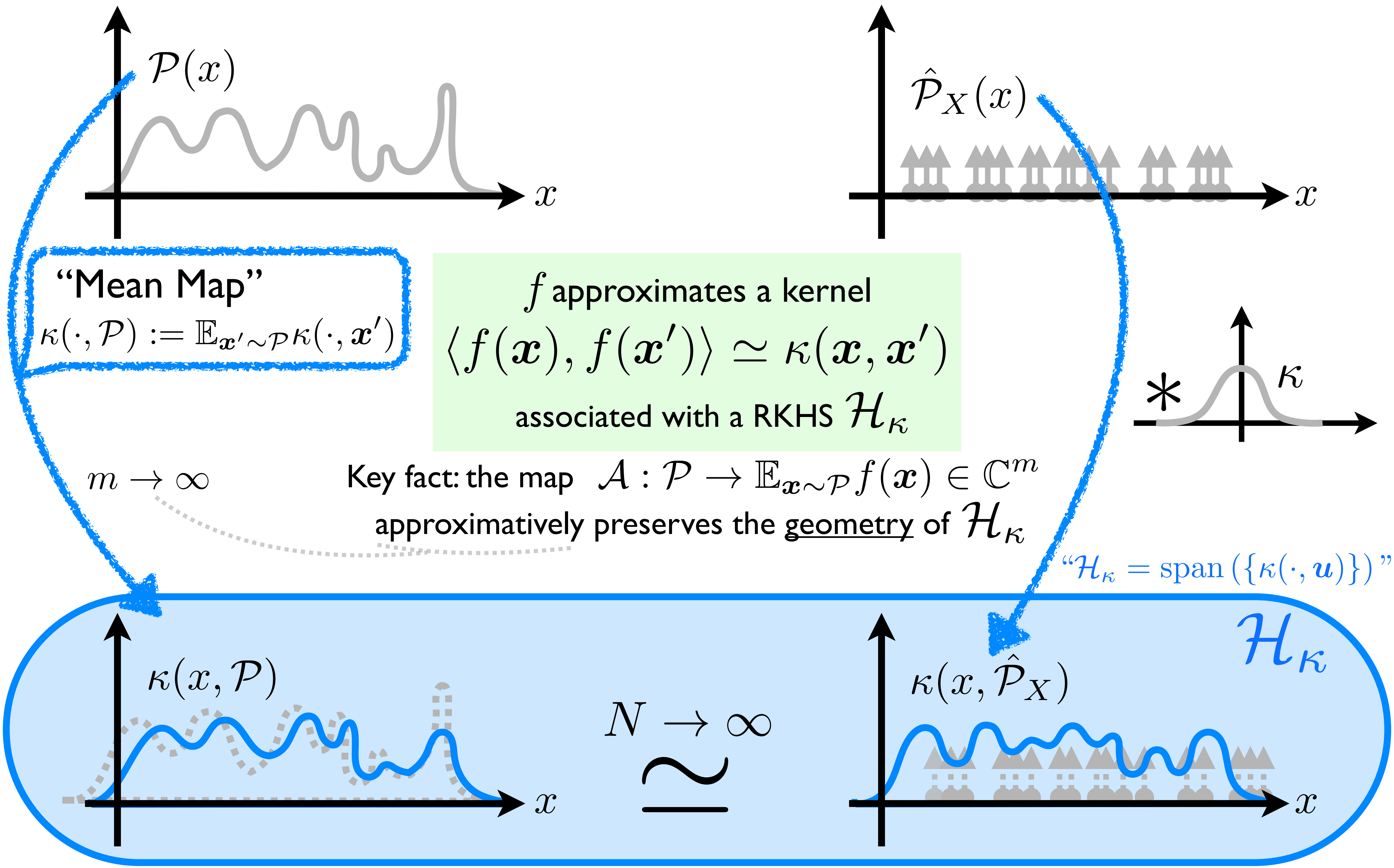
$m \to \infty$

Key fact: the map $\mathcal{A} : \mathcal{P} \to \mathbb{E}_{\boldsymbol{x} \sim \mathcal{P}} f(\boldsymbol{x}) \in \mathbb{C}^m$ approximatively preserves the <u>geometry</u> of $\mathcal{H}_\kappa$

"$\mathcal{H}_\kappa = \mathrm{span}\left(\{\kappa(\cdot, \boldsymbol{u})\}\right)$"

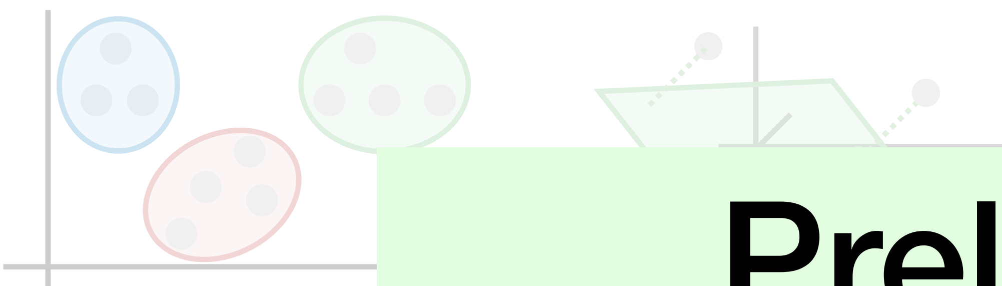


Geometric interpretation



In this talk…

Unsupervised ML

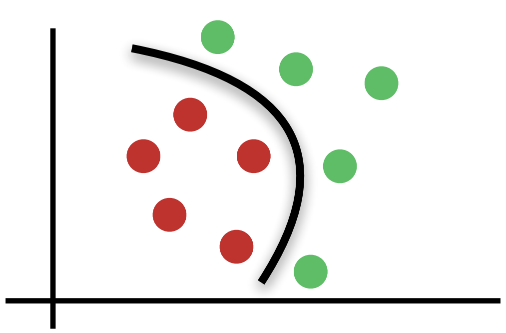


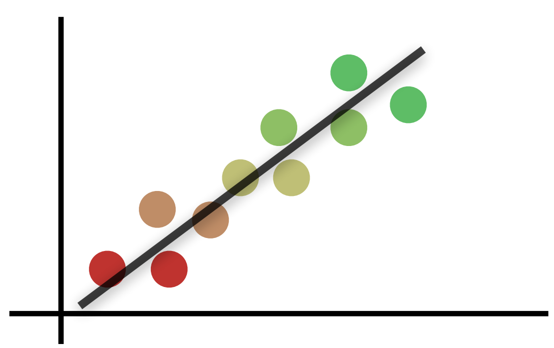
Unsupervised Compressive Learning

- Compressive K-Means [Keriven-CKM]
- n [Keriven-GMM]
- -CL]

Preliminary 2:
(Uncompressed) Classification Basics

Supervised ML



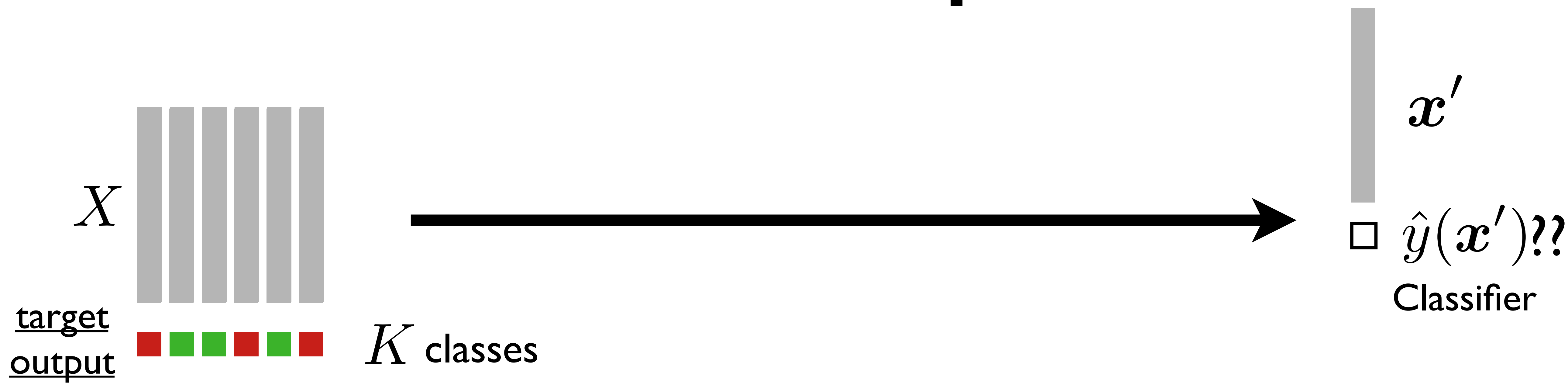


Supervised Compressive Learning

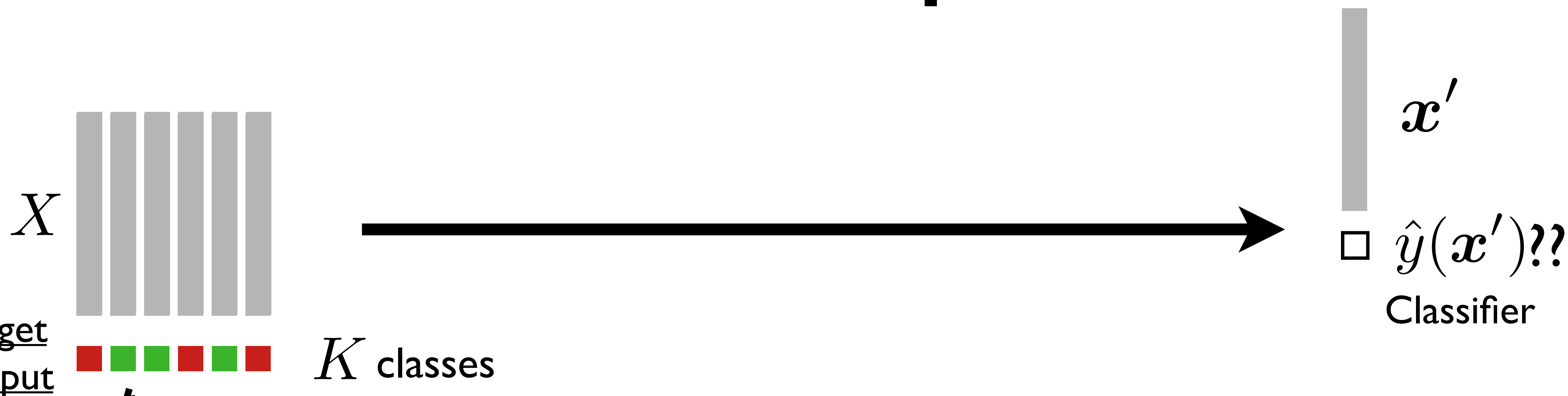
Compressive Classification

(a proof of concept)

Classification problem

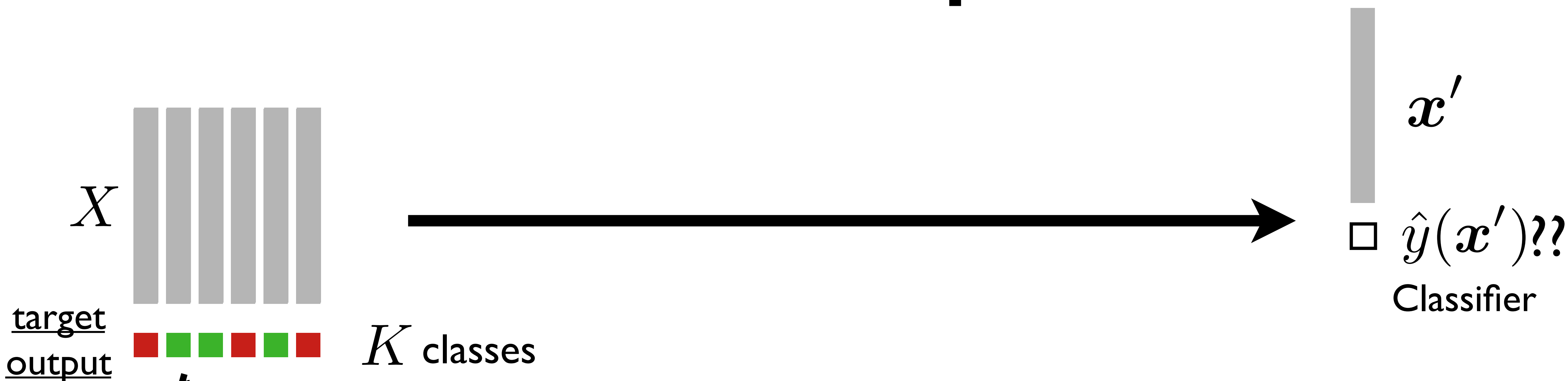


Classification problem



Model: $(\boldsymbol{x}_i, y_i) \sim \sum_{k=1}^{K} p_k \mathcal{P}_k(\boldsymbol{x})$

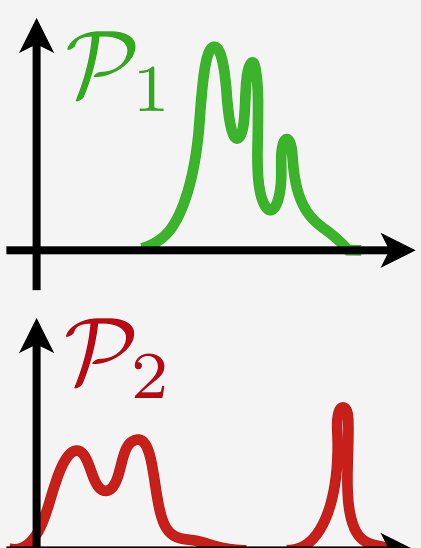
Classification problem



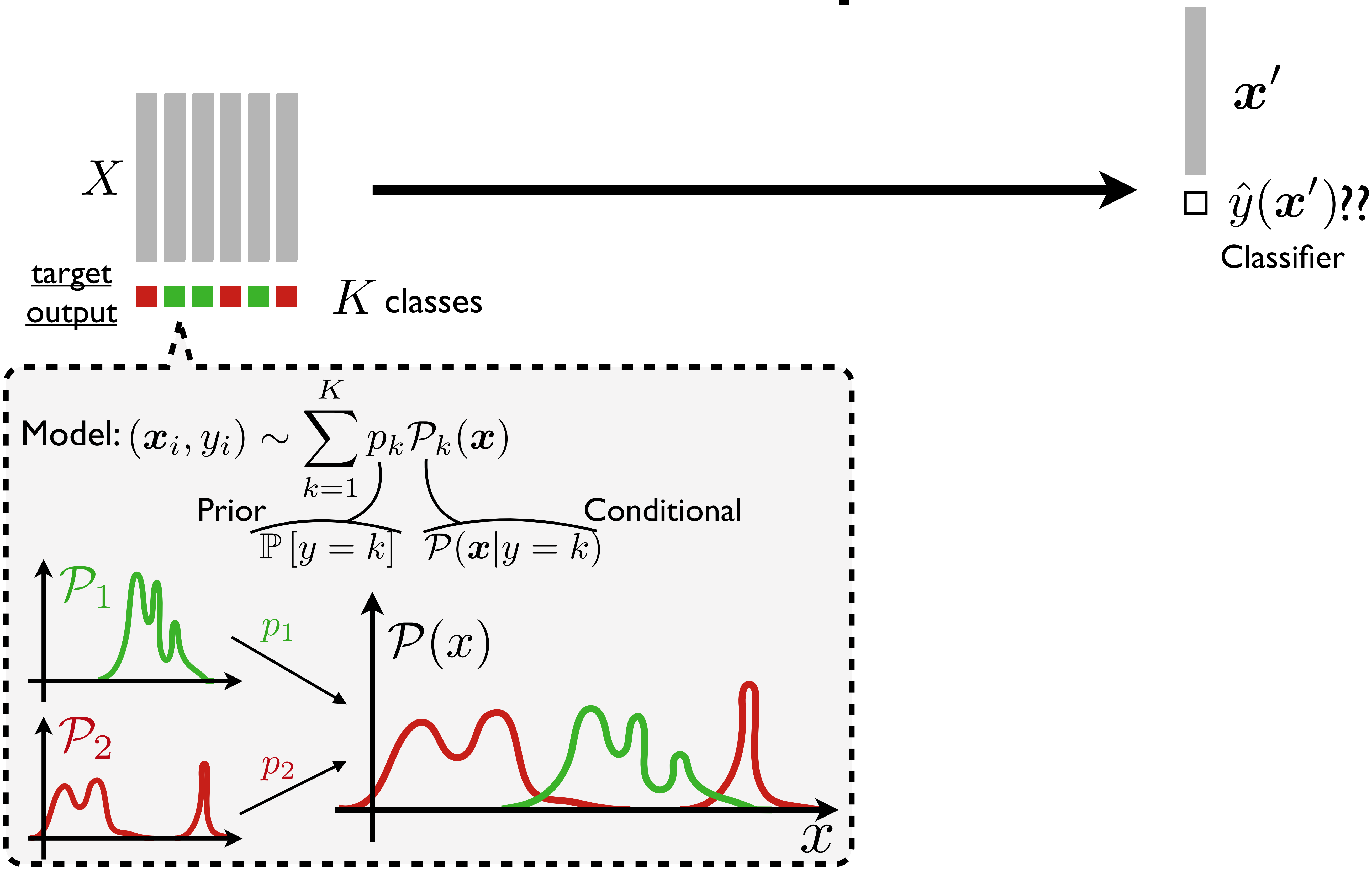
Model: $(\boldsymbol{x}_i, y_i) \sim \sum_{k=1}^{K} p_k \mathcal{P}_k(\boldsymbol{x})$

Conditional

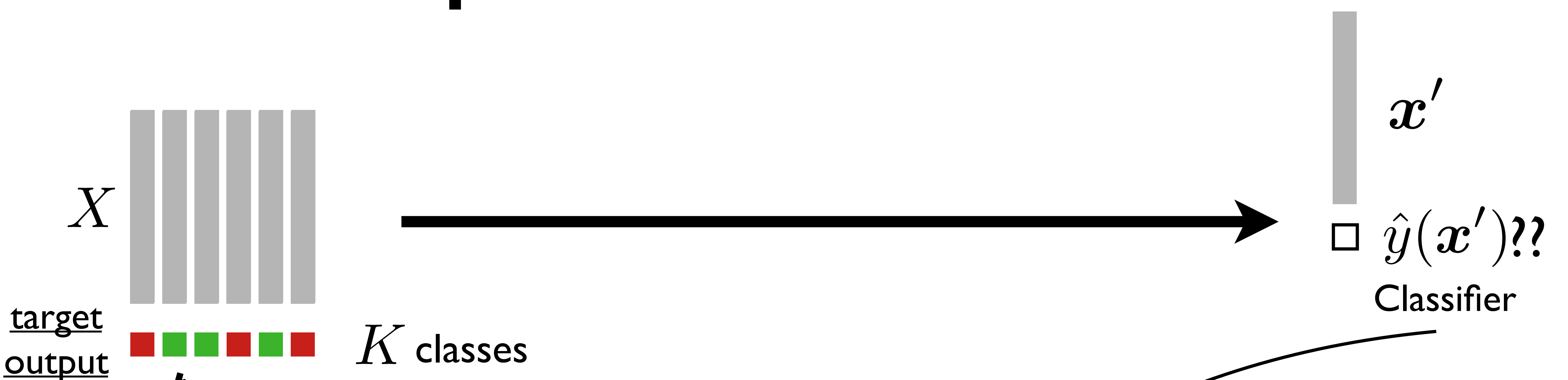
$\mathcal{P}(\boldsymbol{x}|y=k)$



Classification problem

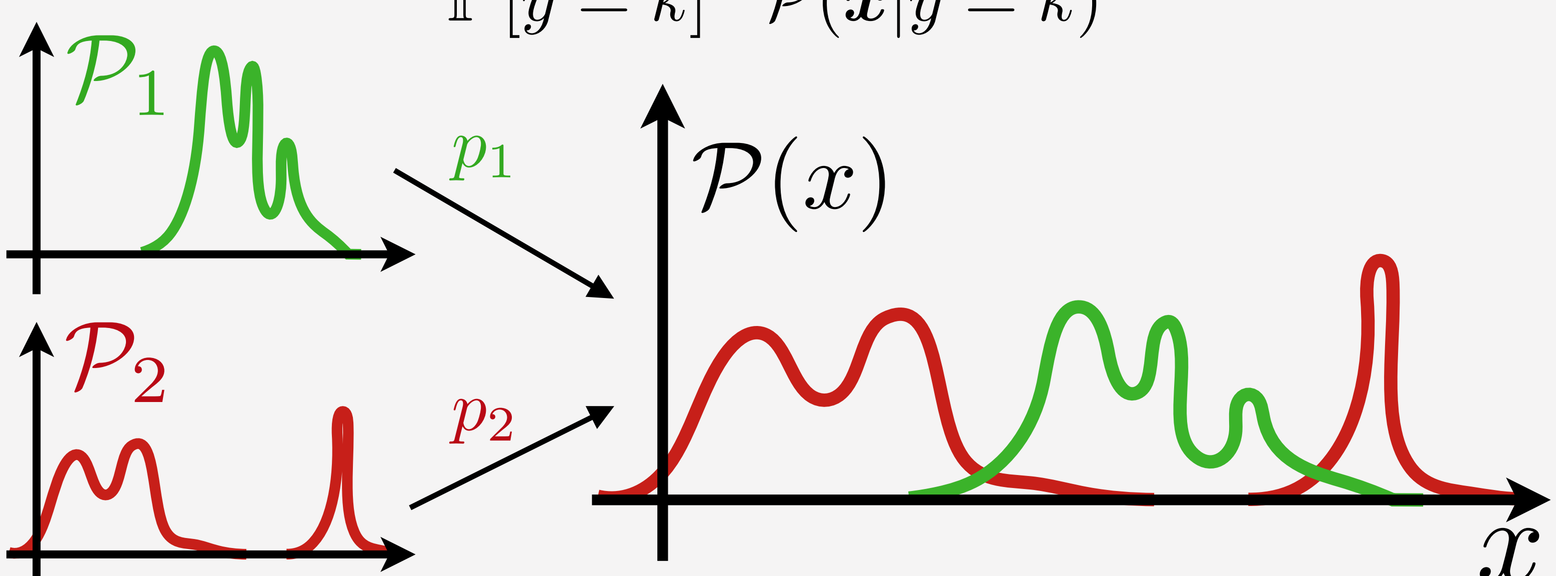


Optimal classifier



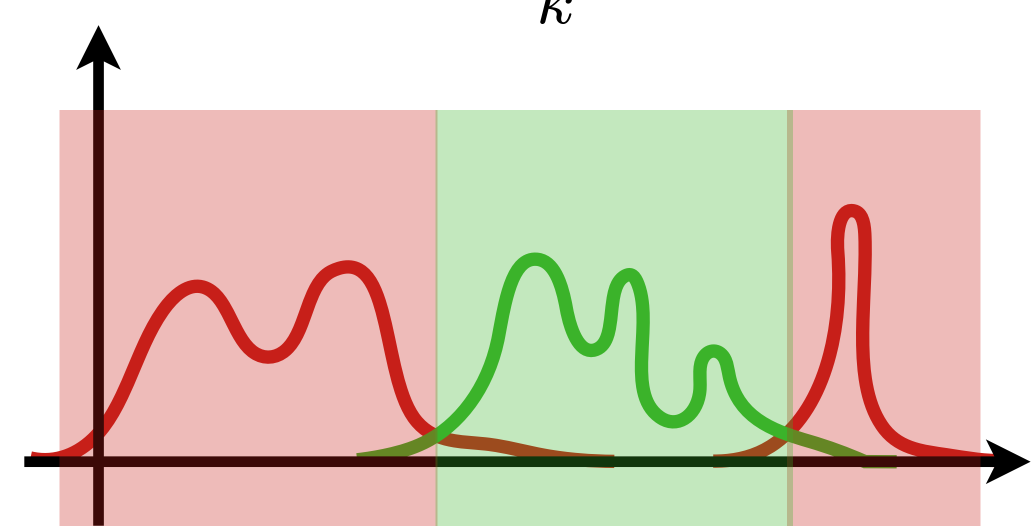
Model: $(\boldsymbol{x}_i, y_i) \sim \sum_{k=1}^{K} \underbrace{p_k}_{\mathbb{P}[y=k]} \underbrace{\mathcal{P}_k(\boldsymbol{x})}_{\mathcal{P}(\boldsymbol{x}|y=k)}$

Prior $\mathbb{P}[y=k]$ Conditional $\mathcal{P}(\boldsymbol{x}|y=k)$

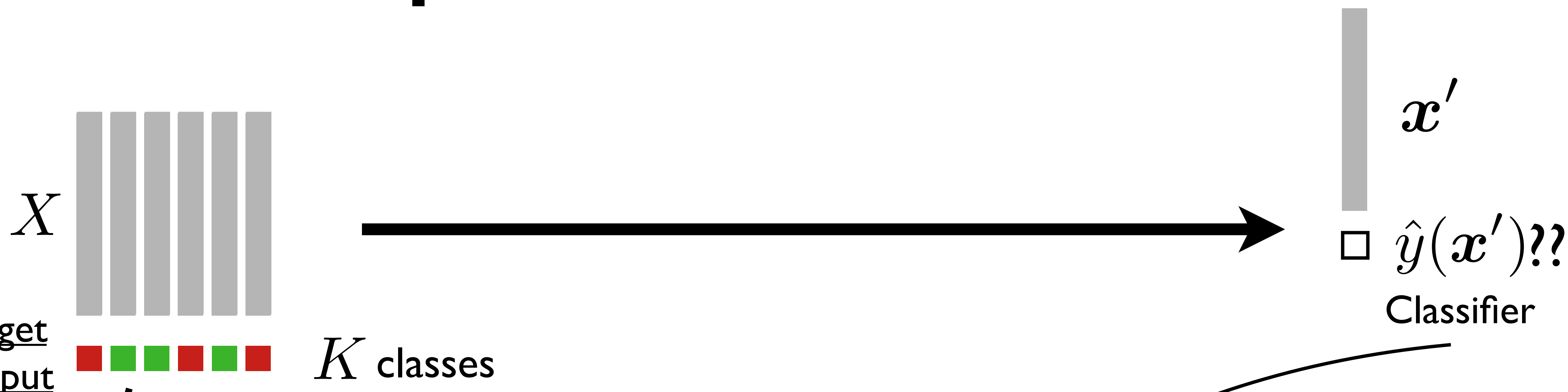


Best we can do (oracle knowledge): MAP (or Bayes) classifier

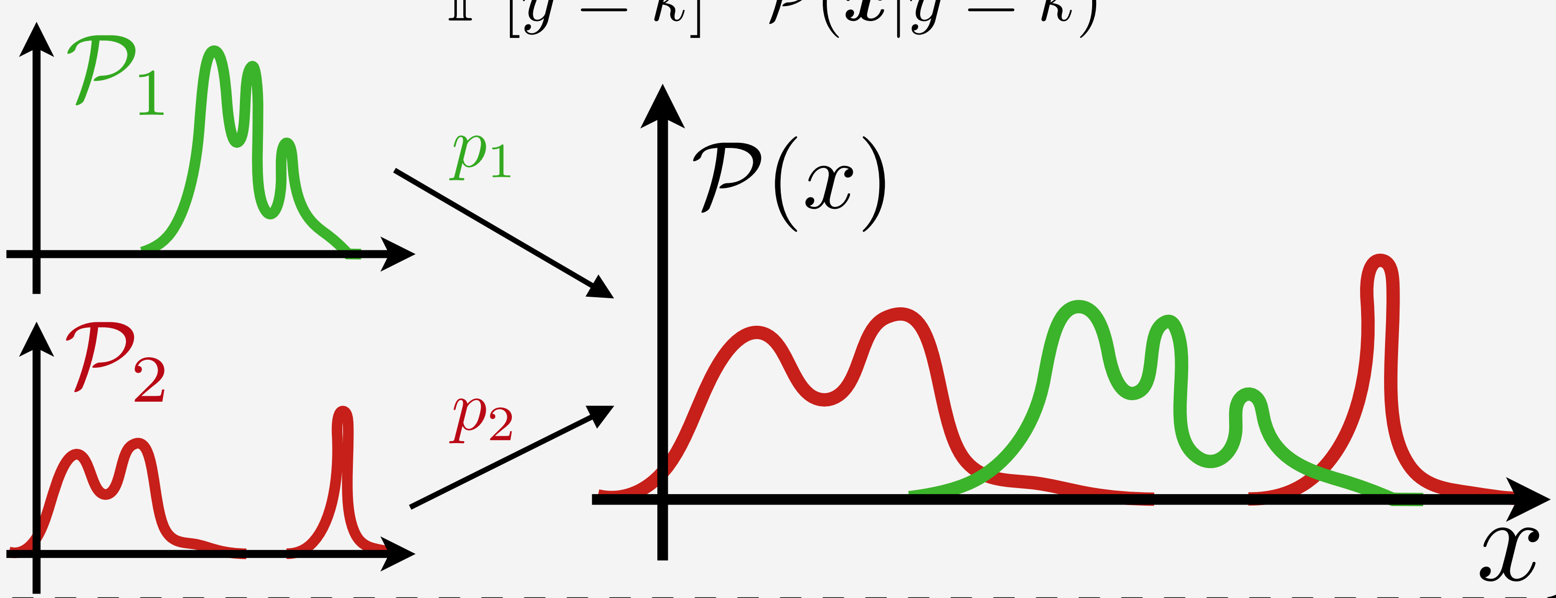
$$\hat{y}^{\mathtt{MAP}} := \arg\max_k p_k \mathcal{P}_k(\boldsymbol{x}')$$



Optimal classifier



Model: $(\boldsymbol{x}_i, y_i) \sim \sum_{k=1}^{K} \underbrace{p_k}_{\text{Prior: } \mathbb{P}[y=k]} \underbrace{\mathcal{P}_k(\boldsymbol{x})}_{\text{Conditional: } \mathcal{P}(\boldsymbol{x}|y=k)}$

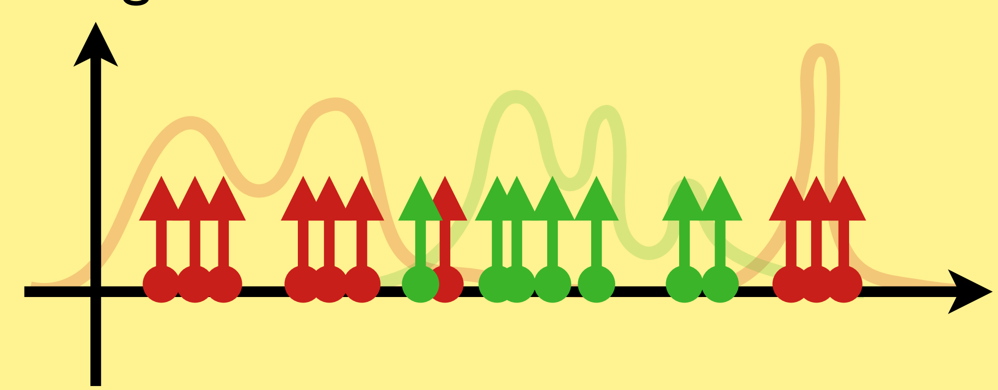


Best we can do (oracle knowledge):
MAP (or Bayes) classifier

$$\hat{y}^{\texttt{MAP}} := \arg\max_k p_k \mathcal{P}_k(\boldsymbol{x}')$$

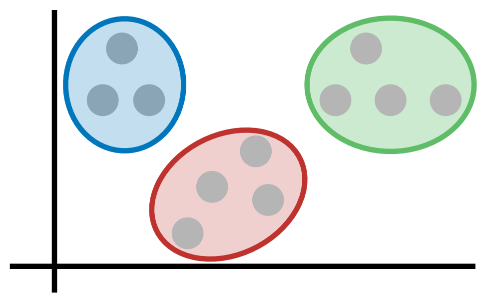
Hard to estimate in practice!

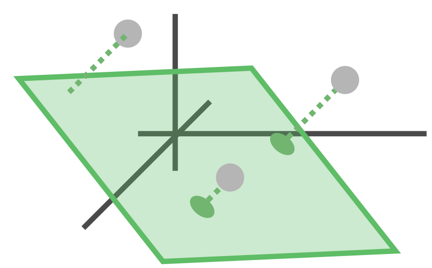
You get this:



In this talk… (finally)

Unsupervised ML

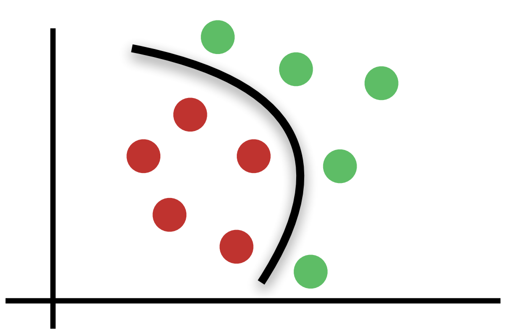


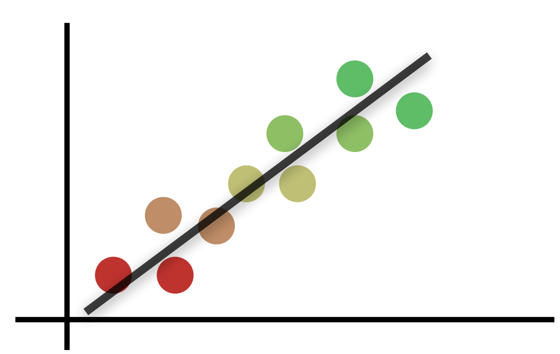


Unsupervised Compressive Learning

- Compressive K-Means [Keriven-CKM]
- Compressive GMM estimation [Keriven-GMM]
- Compressive PCA [Gribonval-CL]

Supervised ML





Supervised Compressive Learning

Compressive Classification

(a proof of concept)

Compressive Classifier

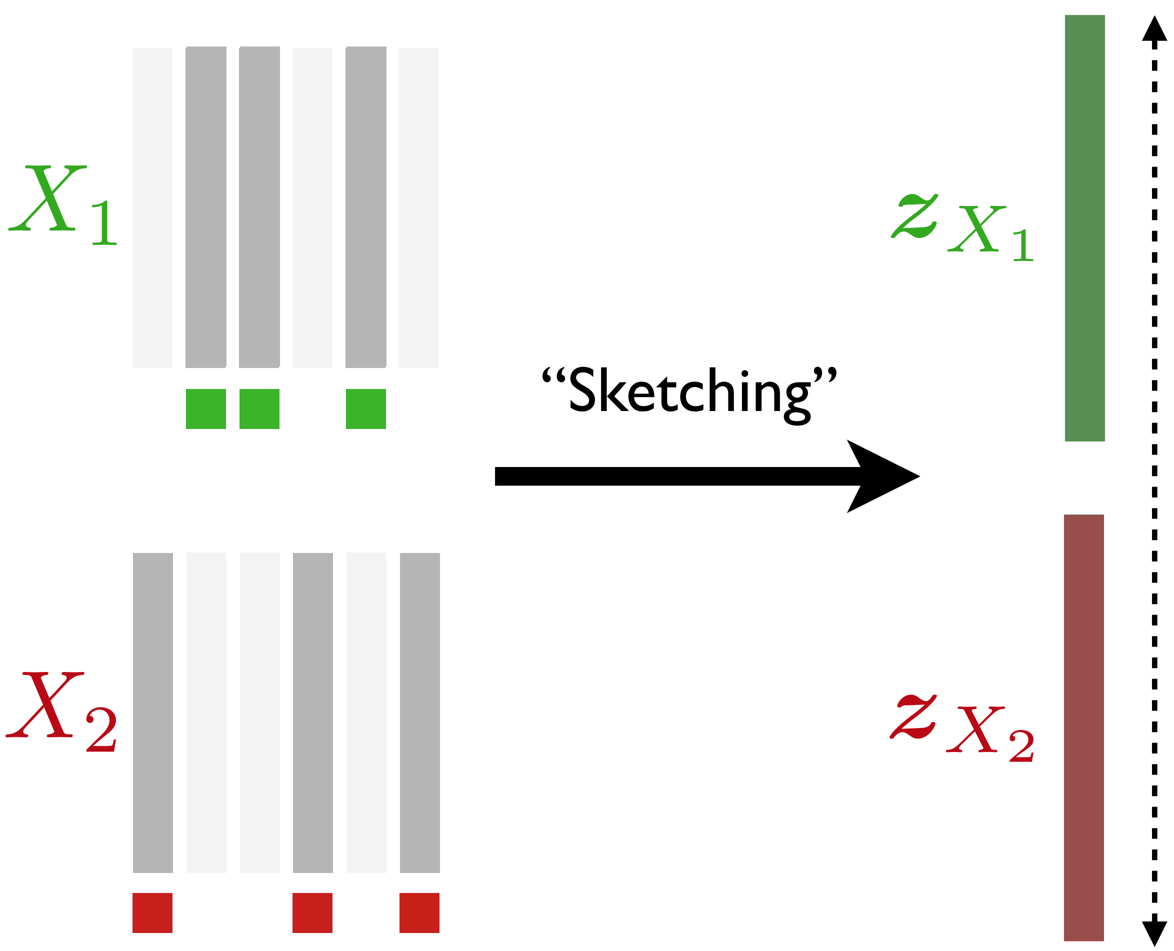
Observation (sketching) phase

One sketch per class! $k \in \{1, \dots, K\}$

$$\boldsymbol{z}_{X_k} = \frac{1}{N_k} \sum_{\boldsymbol{x}_i \in X_k} f(\boldsymbol{x}_i)$$

$$\simeq$$

i.e., sketch the conditionals! $\mathcal{A}(\mathcal{P}_k) = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{P}_k} f(\boldsymbol{x})$



Km coefficients to store

Efficient compression when

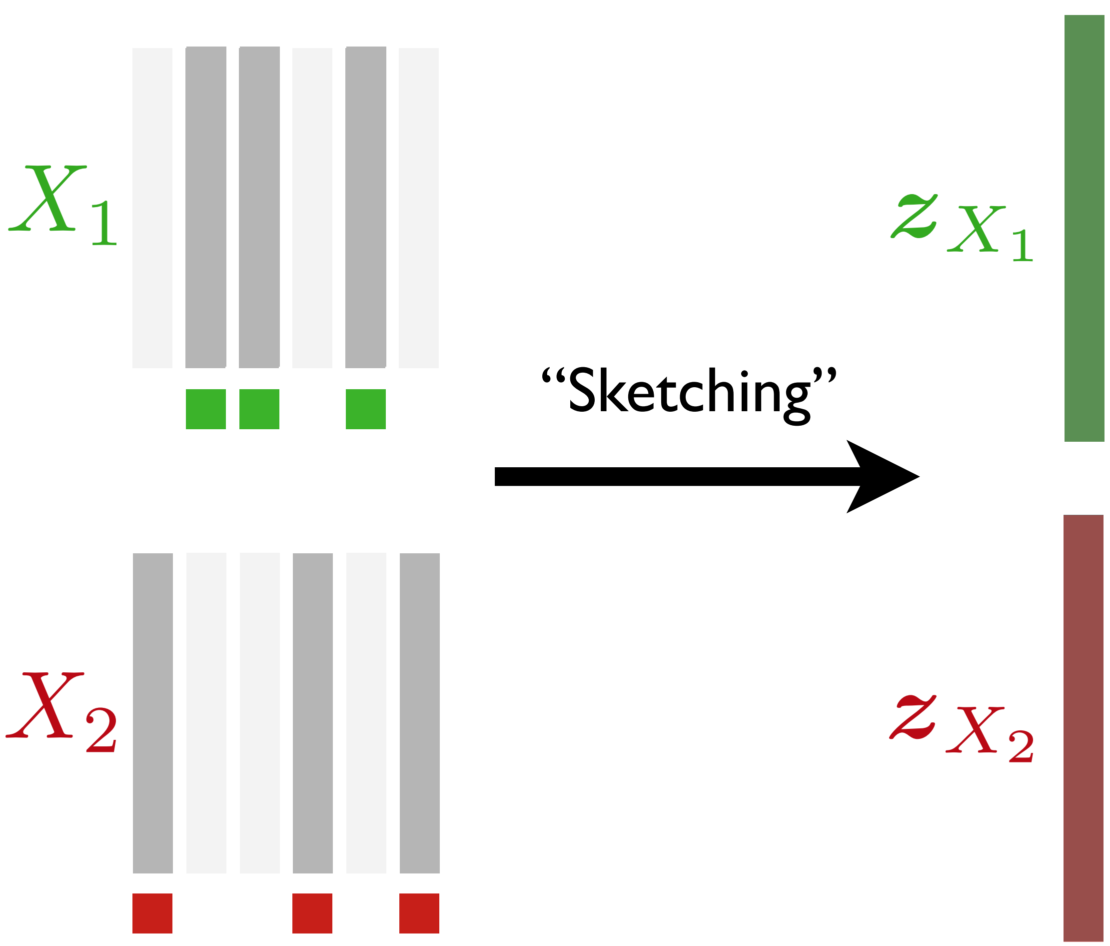
$$Km \ll Nn$$

Compressive Classifier

Observation (sketching) phase

One sketch per class! $\boldsymbol{z}_{X_k} = \frac{1}{N_k} \sum_{\boldsymbol{x}_i \in X_k} f(\boldsymbol{x}_i)$
$k \in \{1, \ldots, K\}$ $\simeq$

i.e., sketch the $\mathcal{A}(\mathcal{P}_k) = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{P}_k} f(\boldsymbol{x})$ conditionals!

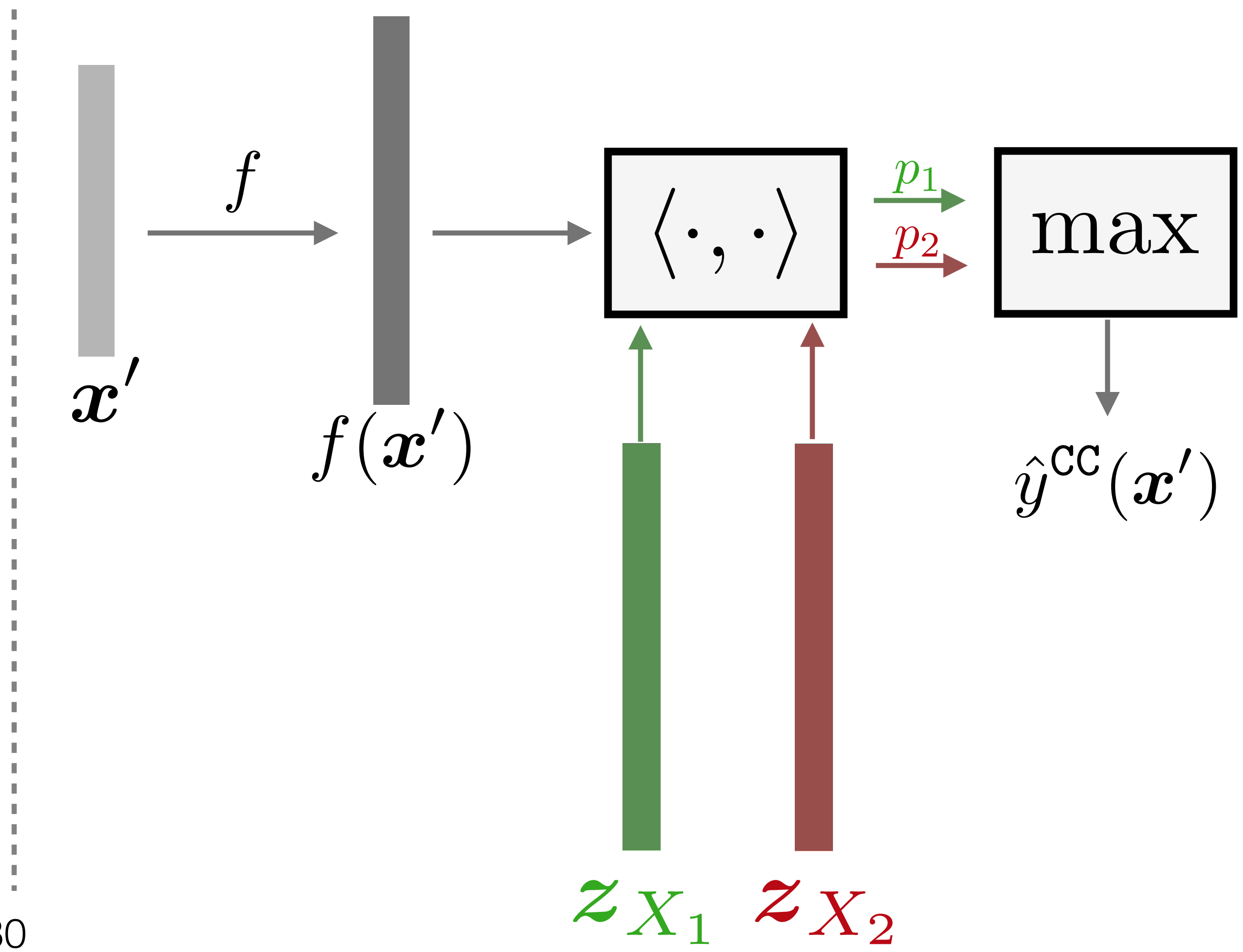


Classification phase

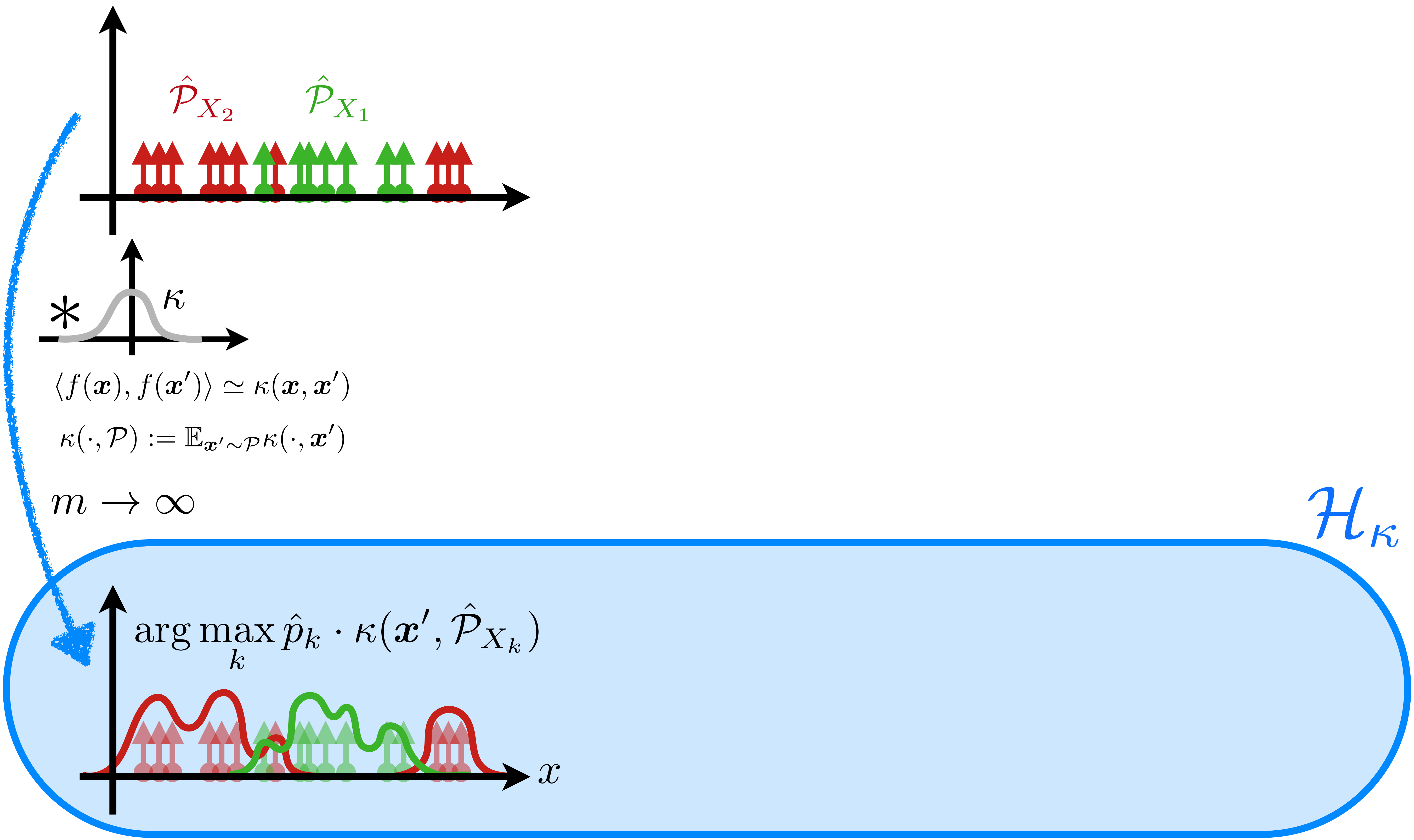
$$\hat{y}^{\mathrm{CC}}(\boldsymbol{x}') := \arg\max_k \hat{p}_k \cdot \langle f(\boldsymbol{x}'), \boldsymbol{z}_{X_k} \rangle$$

$\frac{N_k}{N}$ approximated prior

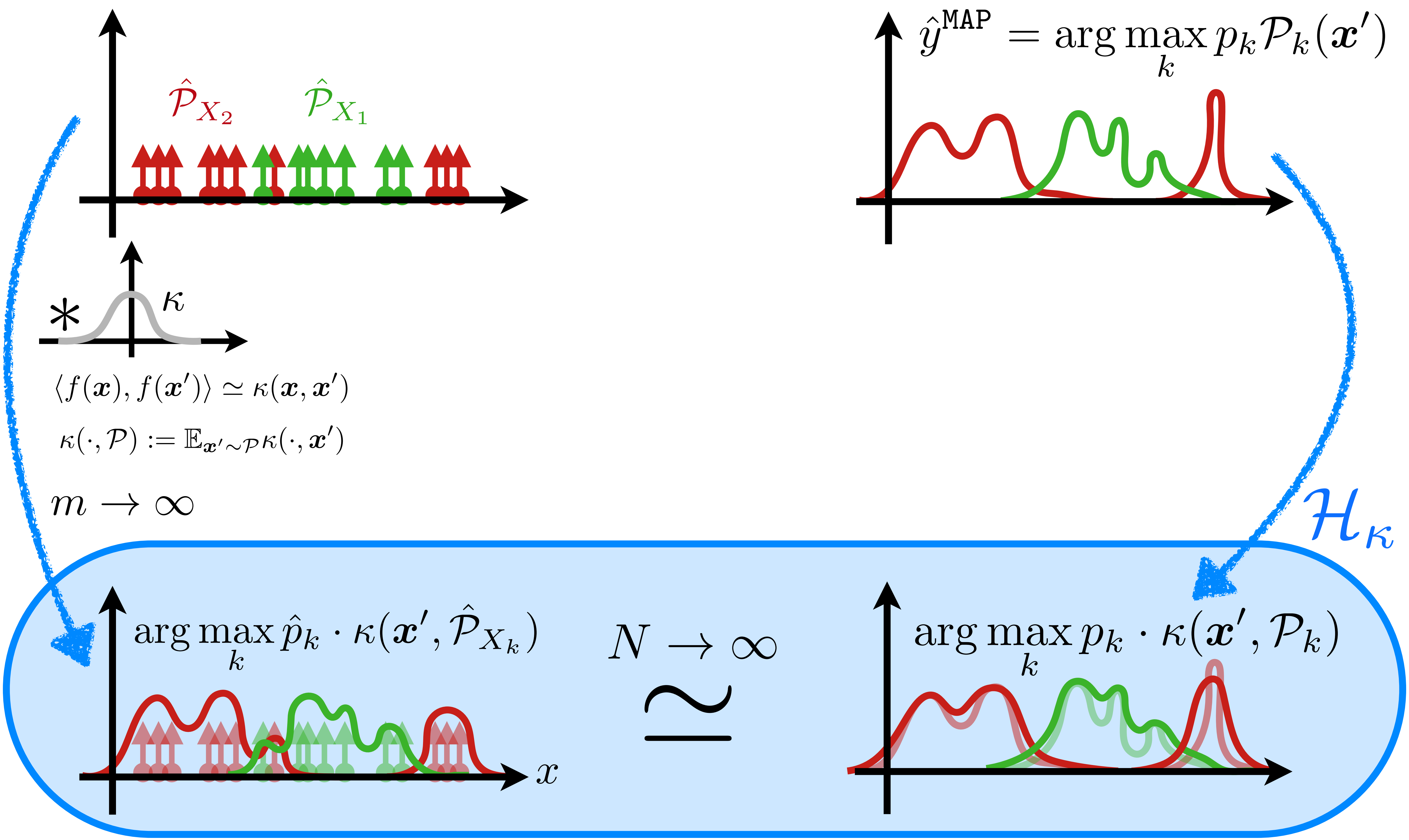
sketch class correlation



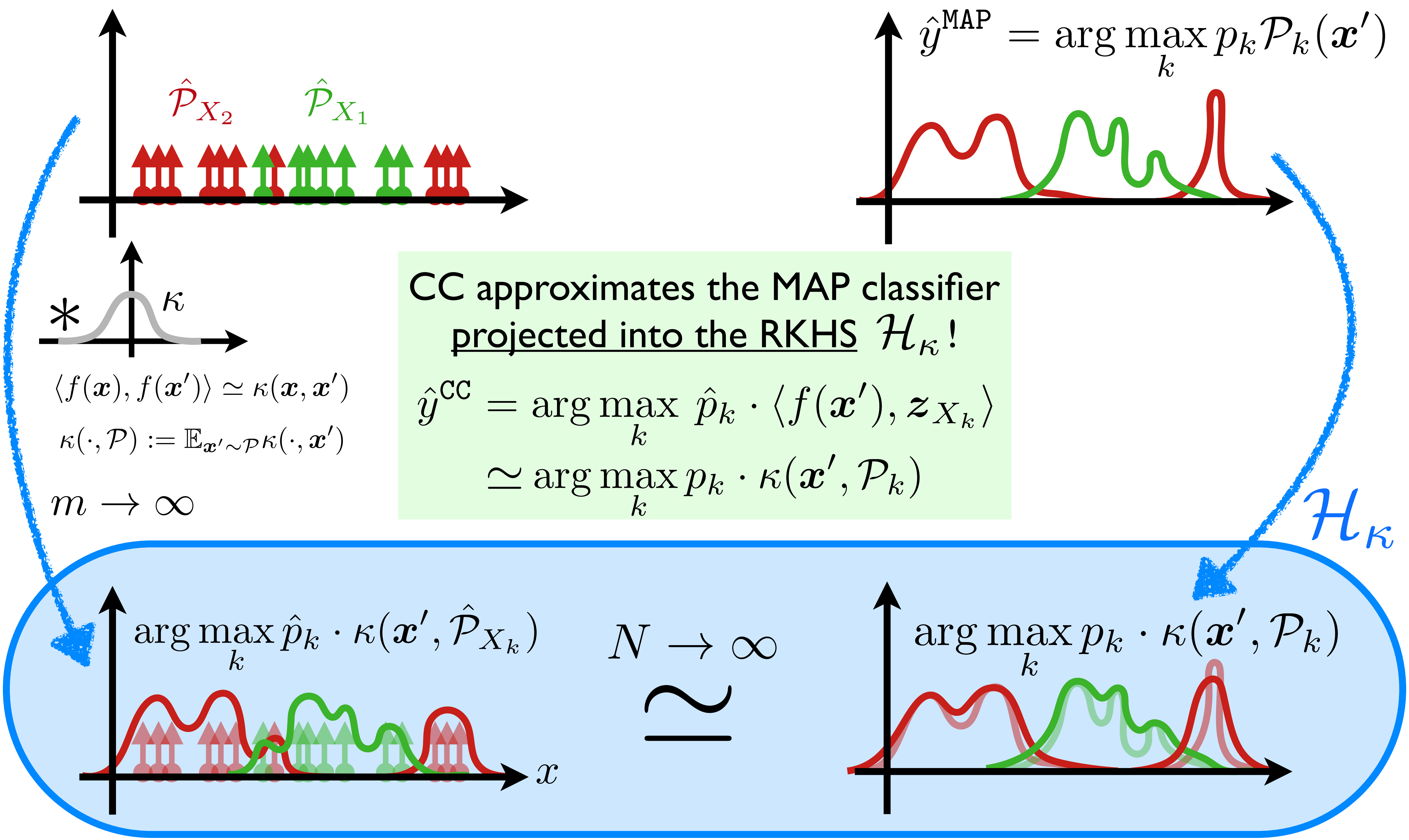
Interpretation



Interpretation



Interpretation



CC approximates the MAP classifier
<u>projected into the RKHS</u> $\mathcal{H}_\kappa$!

$$\hat{y}^{\texttt{CC}} = \arg\max_k \hat{p}_k \cdot \langle f(\boldsymbol{x}'), \boldsymbol{z}_{X_k} \rangle$$
$$\simeq \arg\max_k p_k \cdot \kappa(\boldsymbol{x}', \mathcal{P}_k)$$

Pros and cons

$$\boldsymbol{z}_{X_k} = \frac{1}{N_k} \sum_{\boldsymbol{x}_i \in X_k} f(\boldsymbol{x}_i) \quad \text{then} \quad \begin{aligned} \hat{y}^{\text{CC}} &= \arg\max_k \hat{p}_k \cdot \langle f(\boldsymbol{x}'), \boldsymbol{z}_{X_k} \rangle \\ &\simeq \arg\max_k p_k \cdot \kappa(\boldsymbol{x}', \mathcal{P}_k) \end{aligned}$$

Pro

- Cheap to "learn": you only observe the dataset ("compressive k-NN")
- Cheap to evaluate: f (once), +, $*$ and max
- Easy to parallelize/update: suited to massive datasets, distributed (sensitive?) datasets, data streams, and data augmentation (almost free!)
- MAP in RKHS interpretation is seducing, but it is an asymptotical result…

Pros and cons

$$\boldsymbol{z}_{X_k} = \frac{1}{N_k} \sum_{\boldsymbol{x}_i \in X_k} f(\boldsymbol{x}_i) \quad \text{then} \quad \begin{aligned} \hat{y}^{\mathsf{CC}} &= \arg\max_k \; \hat{p}_k \cdot \langle f(\boldsymbol{x}'), \boldsymbol{z}_{X_k} \rangle \\ &\simeq \arg\max_k p_k \cdot \kappa(\boldsymbol{x}', \mathcal{P}_k) \end{aligned}$$

Pro

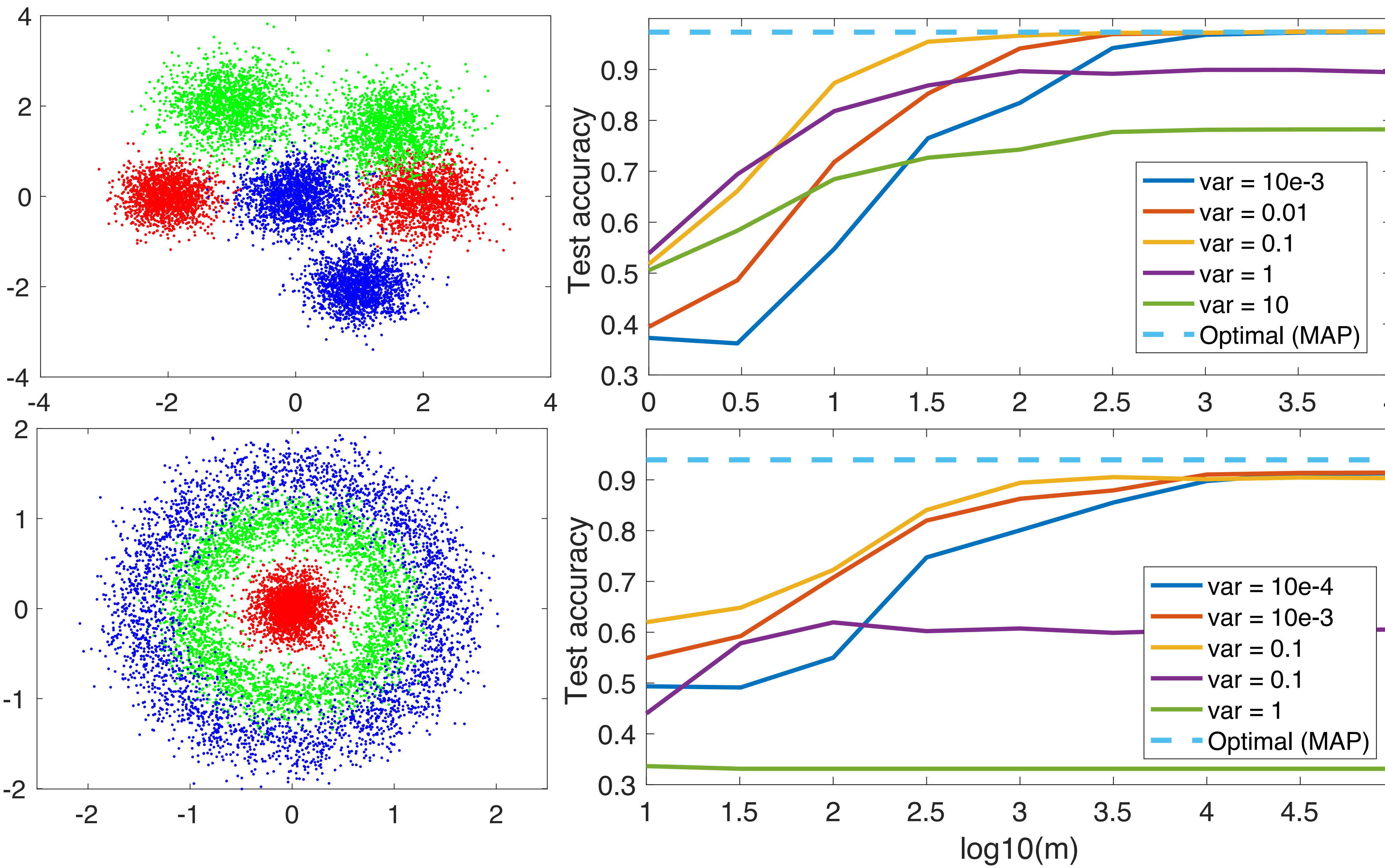
- Cheap to "learn": you only observe the dataset ("compressive k-NN")
- Cheap to evaluate: f (once), +, $*$ and max
- Easy to parallelize/update: suited to massive datasets, distributed (sensitive?) datasets, data streams, and data augmentation (almost free!)
- MAP in RKHS interpretation is seducing, but it is an asymptotical result…

Con

- No guarantees (yet): generalization error bounds, sample complexity bounds…
- Hyperparameters: choice of f (hence κ) is crucial!
- High-Dimensional data is expected to be difficult

Proof of concept (synthetical)

Random Fourier Features sketch with Gaussian kernel



Proof of concept (real)

Random Fourier Features sketch with Gaussian kernel

Standard learned classifier

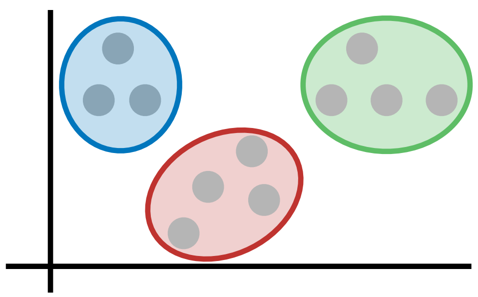
Compressive classifier

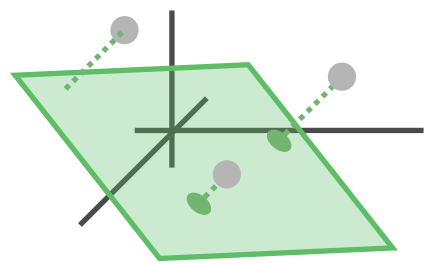
	N	n	K	SVM	$m = 50$	$m = 1000$
Iris	150	4	3	2.00	6.51 ± 1.81	5.51 ± 1.23
				4.00	8.22 ± 3.25	6.18 ± 2.40
Wine	178	13	3	0.84	4.56 ± 2.34	2.43 ± 0.72
				1.69	13.75 ± 4.09	8.19 ± 1.29
Breast cancer	569	30	2	3.67	7.00 ± 1.40	3.93 ± 0.39
				2.13	9.22 ± 2.33	6.23 ± 0.69
Adult (3 attr.)	30718	3	2	21.03	23.88 ± 4.37	23.11 ± 1.05
				21.06	36.09 ± 6.67	35.04 ± 1.63

Error rates (%) with SD
on train and test sets

In this talk…

Unsupervised ML

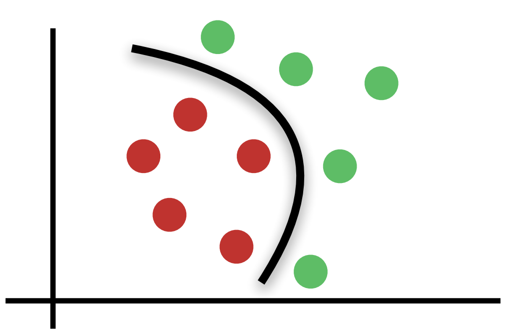


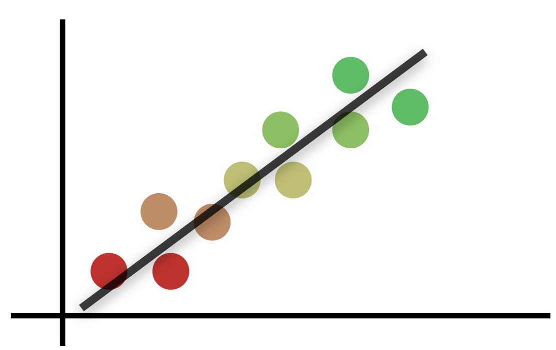


Unsupervised Compressive Learning

- Compressive K-Means [Keriven-CKM]
- Compressive GMM estimation [Keriven-GMM]
- Compressive PCA [Gribonval-CL]

Supervised ML





Supervised Compressive Learning

Compressive Classification

(a proof of concept)

What about HD data such as images?

Sketching images

Up to now, the sketch function f used are Random Fourier Features

$$f(\boldsymbol{x})_{\mathtt{RFF}} = \left[\exp(\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x})\right]_{j=1}^{m}$$

$$\Updownarrow$$

$\kappa(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{x} - \boldsymbol{x}')$ Depends <u>only</u> on signal difference

For images, a kernel based on the pixel-wise difference does not make so much sense…

Idea: capture the image structure with a (random) Convolutional Neural Network architecture

$$f(\boldsymbol{x})_{\mathtt{CNN}} = \mathtt{CNN}_{\theta}(\boldsymbol{x})$$

Random CNN have been shown to be surprisingly meaningful features for image processing tasks

E.g., [Cho], [Giryes], [Rosenfeld], …

Sketching images

Up to now, the sketch function f used are Random Fourier Features

$$f(\boldsymbol{x})_{\texttt{RFF}} = \left[\exp(\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x})\right]_{j=1}^{m}$$

$$\Updownarrow$$

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{x} - \boldsymbol{x}')$$

For images, a kernel based on the pixel-wise difference does not make much sense…

Idea: capture the image structure with a (random) Convolutional Neural Network architecture

$$f(\boldsymbol{x})_{\texttt{CNN}} = \texttt{CNN}_{\theta}(\boldsymbol{x})$$

	N	n	CNN	m = 250	m = 5000
MNIST	60000	$28 \times 28 \times 1$	1.60 ± 0.12	17.73 ± 1.43	16.60 ± 1.54
	10000		1.63 ± 0.11	16.83 ± 1.39	15.80 ± 1.61
CIFAR10	50000	$32 \times 32 \times 3$	39.08 ± 1.48	71.76 ± 1.85	72.83 ± 2.00
	10000		40.28 ± 1.36	71.12 ± 1.72	72.02 ± 1.85

"Not that random" (Work In Progress)…

Some perspectives

$$\boldsymbol{z}_{X_k} = \frac{1}{N_k} \sum_{\boldsymbol{x}_i \in X_k} f(\boldsymbol{x}_i) \quad \text{then} \quad \begin{aligned} \hat{y}^{\mathsf{CC}} &= \arg\max_k \ \hat{p}_k \cdot \langle f(\boldsymbol{x}'), \boldsymbol{z}_{X_k} \rangle \\ &\simeq \arg\max_k p_k \cdot \kappa(\boldsymbol{x}', \mathcal{P}_k) \end{aligned}$$

- Explore the features/kernel choice?
- Learn the kernel from a small data sample (distilled sensing)?
- (Non-asymptotical) formal guarantees?
- Is this the best we can do with the sketch?
- Regression? Other supervised tasks?
- …

(Some) references

- [Keriven-CKM] N. Keriven, N. Tremblay, Y. Traonmilin, R. Gribonval, "Compressive K-Means", ICASSP 2017
- [Keriven-GMM] N. Keriven, A. Bourrier, R. Gribonval, P. Perez, "Sketching for Large-Scale Learning of Mixture Models", Information and Inference, 2017
- [Gribonval-CL] R. Gribonval, G. Blanchard, N. Keriven, Y. Traonmilin, "Compressive Statistical Learning with Random Feature Moments", ArxiV, 2017
- [Giryes] R. Giryes, G. Sapiro, A.M. Bronstein, "Deep Neural Networks with Random Gaussian Weights: A Universal Classification Strategy?", IEEE Transactions on Signal Processing, 2016
- [Cho] Y. Cho, L.K. Saul, "Kernel methods for deep learning", NIPS 2009
- [Rosenfeld] A. Rosenfeld, J.K. Tsotsos, "Intriguing Properties of Randomly Weighted Networks: Generalizing While Learning Next to Nothing", ArxiV, 2018
- [Smola] A. Smola, A. Gretton, L. Song, B. Scholkopf, "A Hilbert space embedding for distributions", International Conference on Algorithmic Theory, 2007