SENSING-AWARE CLASSIFICATION WITH HIGH DIMENSIONAL DATA

Prakash Ishwar

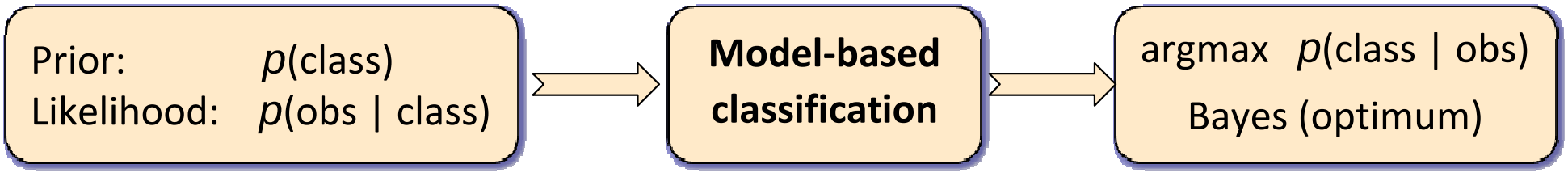
Burkay Orten, W. Clem Karl, Venkatesh Saligrama, Homer Pien

Classification

- **Model-based classification**

Given: Model

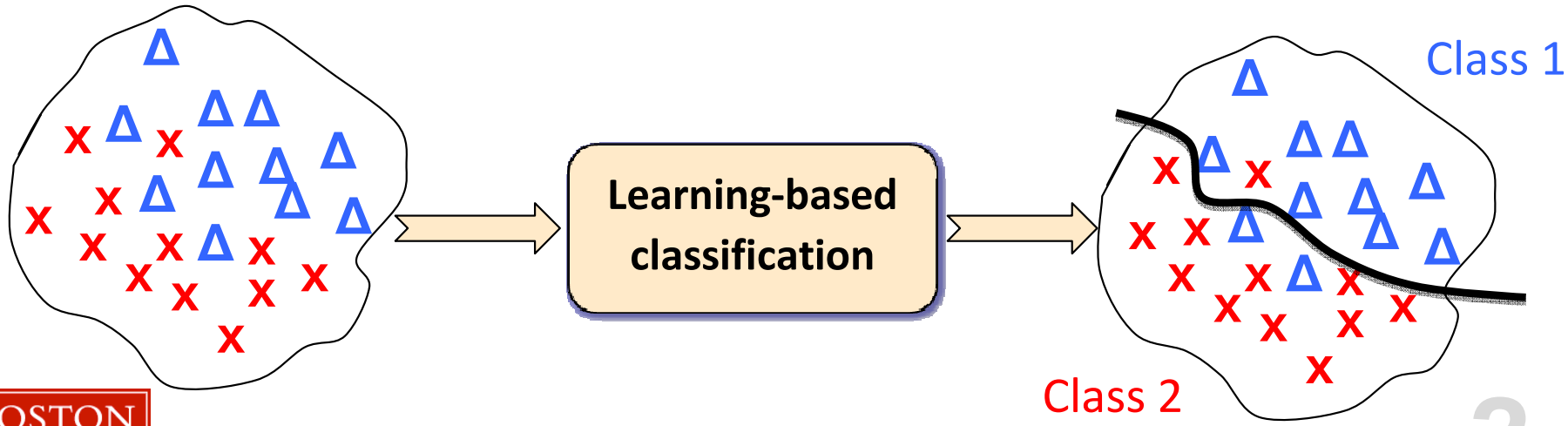
Decision Rule



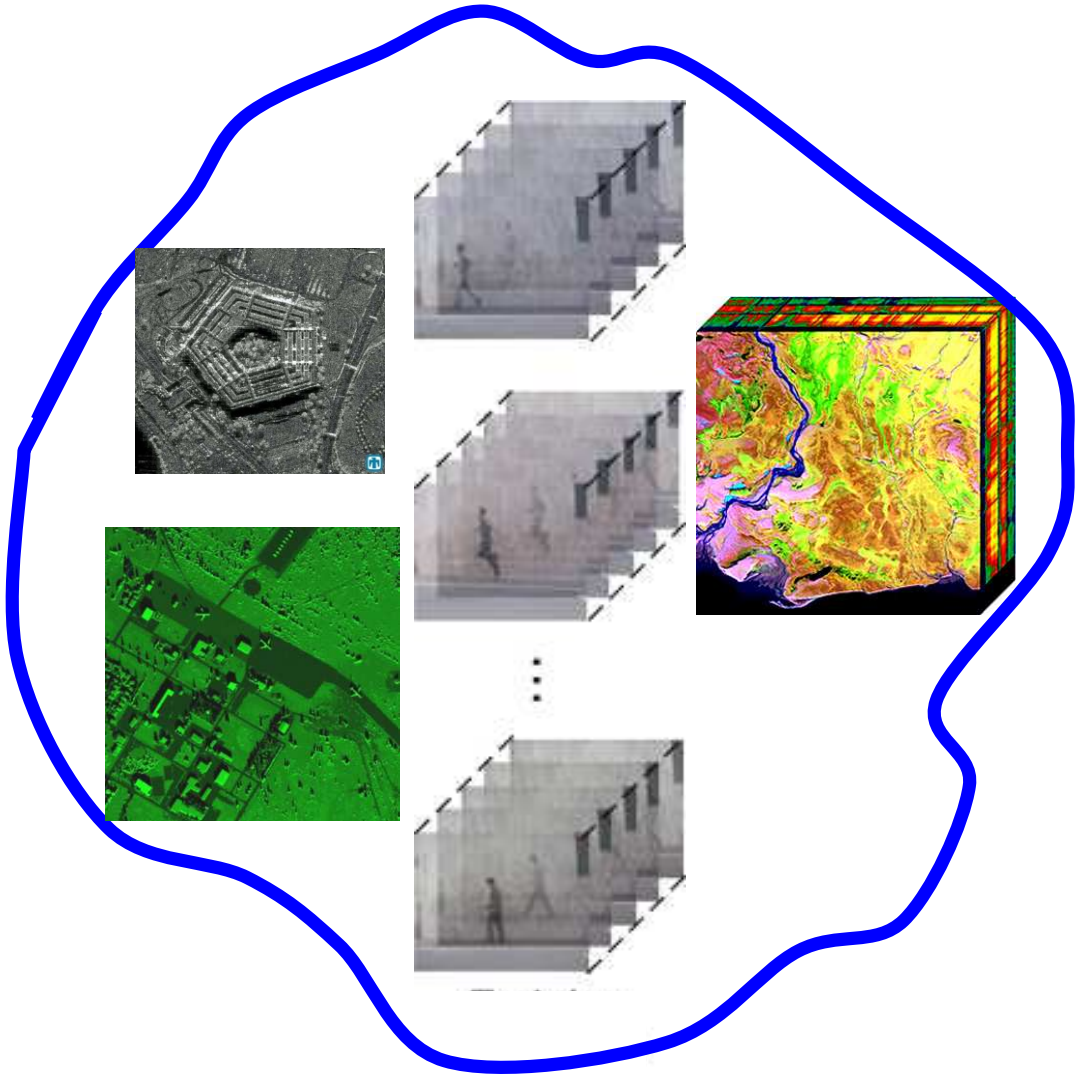
- **Learning-based classification**

Given: Labeled Examples

Decision Rule



Challenges and Approach



Challenges:

1. **High dimensional data**
   - Surveillance video
   - Hyperspectral images
   - SAR
   - MRI…
2. **Few samples**

Approach:

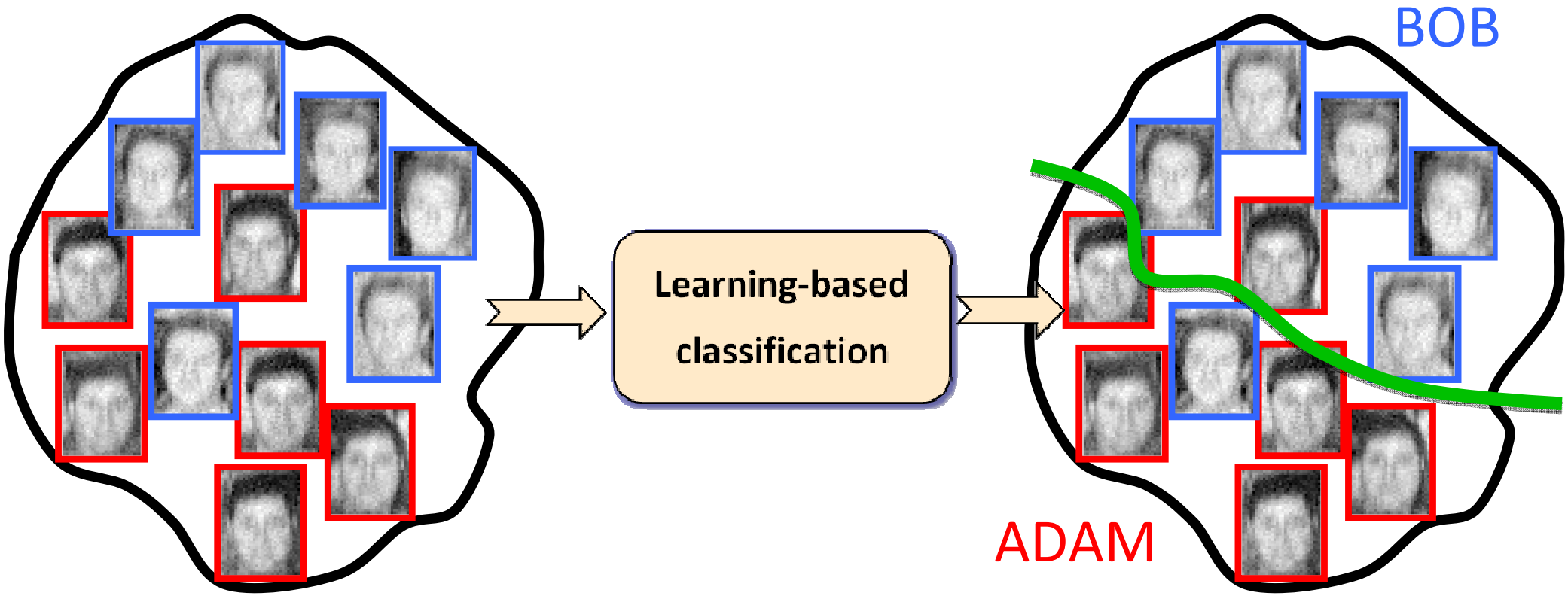
- Exploit latent low-dimensional sensing structure

BOSTON UNIVERSITY

"Sensing-Unaware" Classification

Given: Blurry and noisy faces

Decision Rule



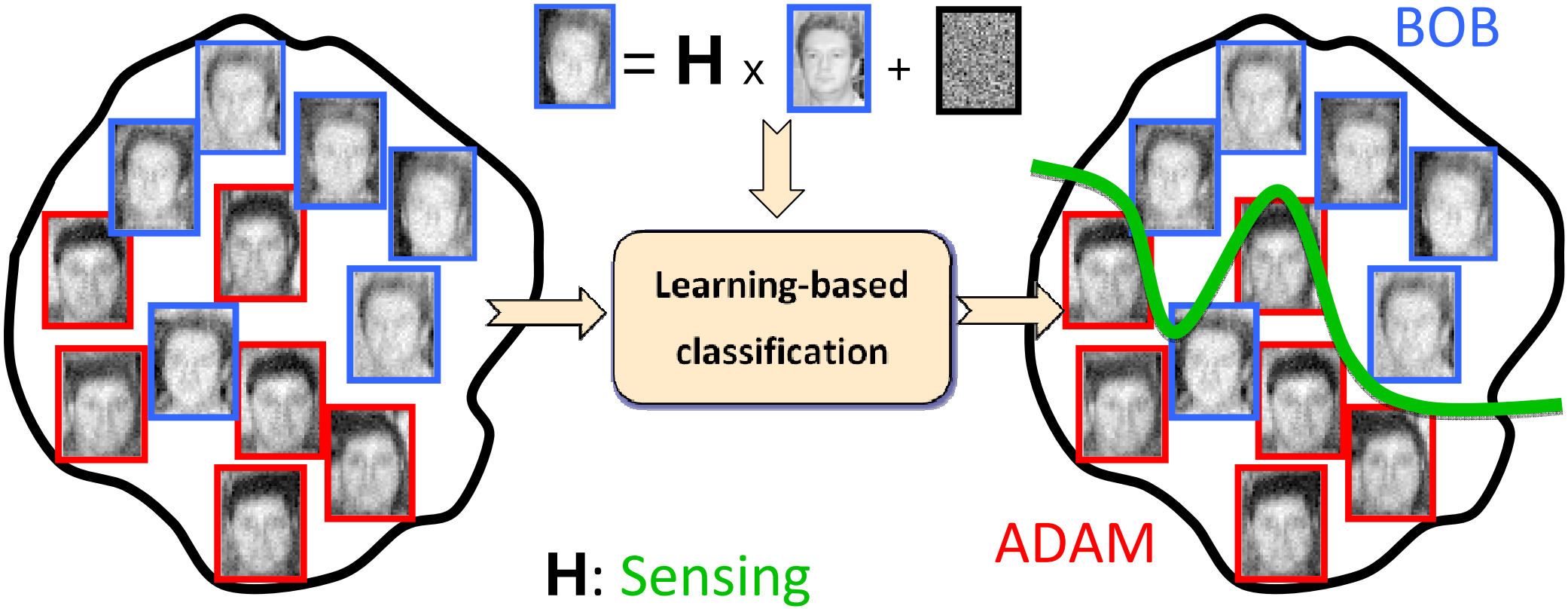
"Sensing-Aware" Classification

Given: Blurry and noisy faces

&

Sensing structure

Decision Rule



H: Sensing Matrix

BOSTON UNIVERSITY

Asymptotic classification performance

- **Asymptotic regime of interest:**
  1. Data-dimension, # samples $\longrightarrow \infty$
  2. Samples per dimension $\longrightarrow 0$
  3. Fixed problem difficulty: Bayes risk kept fixed as dimension scales, i.e., problem not asymptotically "easy".

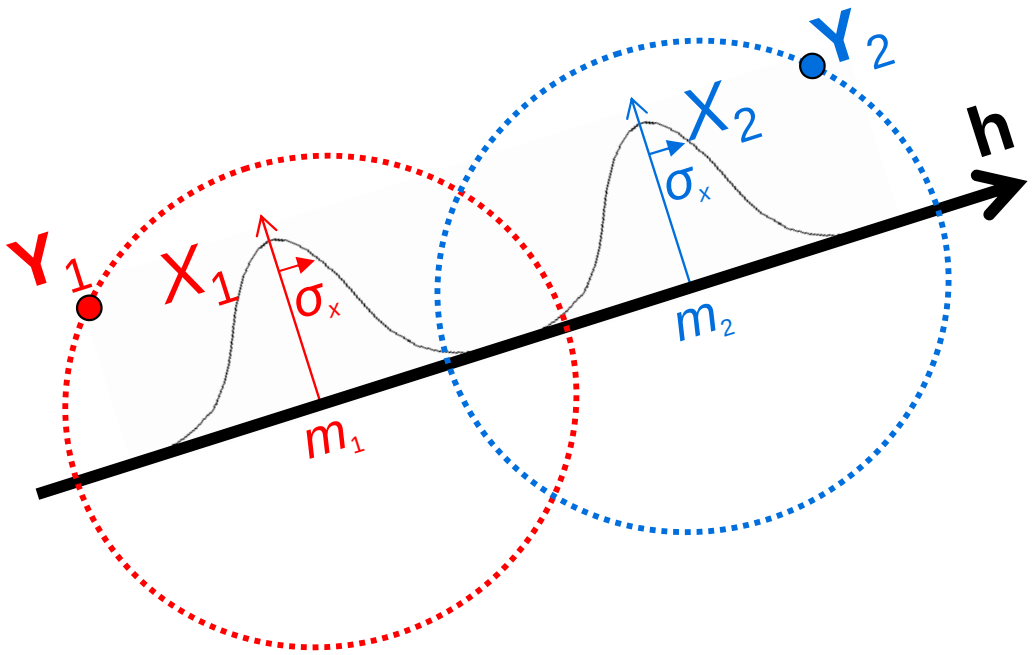
- **Fundamental issue**: asymptotic classification performance
  - $P_{\text{error}} \longrightarrow 1/2$ (random guessing)?
  - $P_{\text{error}} \longrightarrow P_{\text{Bayes}}$?

Mathematical model

- Sensing model:

$$\mathbf{Y}_i = \mathbf{h}X_i + \mathbf{Z}_i$$

$$\begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1} = \begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1} X_i + \begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1}$$



i = 1, 2 (class index)
equally likely

- $X_i \sim \mathcal{N}(m_i, \sigma_x^2), \quad \mathbf{Z}_i \sim \mathcal{N}(\mathbf{0}, \sigma_z^2\mathbf{I})$

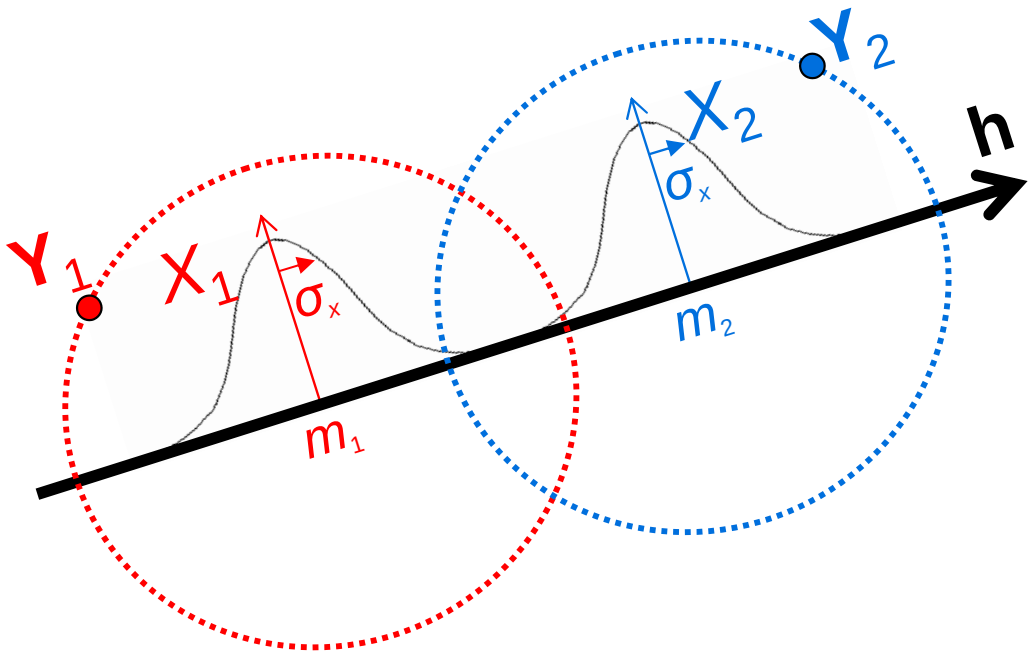
BOSTON UNIVERSITY

Mathematical model

- Sensing model:

$$\mathbf{Y}_i = \mathbf{h}X_i + \mathbf{Z}_i$$

$$\begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1} = \begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1} X_i + \begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1}$$



i = 1, 2 (class index), equally likely

- Given: *n* labeled iid training examples, design classifier

$$\begin{pmatrix} \mathbf{Y}_{11}, \mathbf{Y}_{12}, \ldots, \mathbf{Y}_{1n} \\ \mathbf{Y}_{21}, \mathbf{Y}_{22}, \ldots, \mathbf{Y}_{2n} \end{pmatrix}$$

← Class 1 training examples

← Class 2 training examples

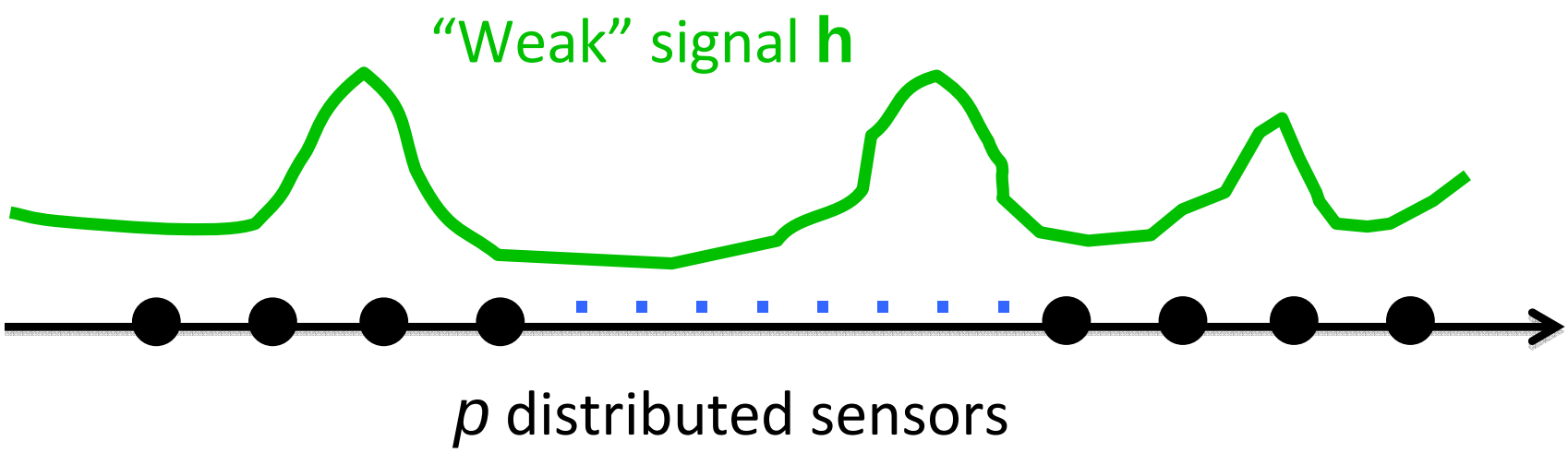
BOSTON UNIVERSITY

A sensor network application

Simple observation model using distributed sensors:

Class 1: **Y** = **h** + noise

Class 2: **Y** = -**h** + noise

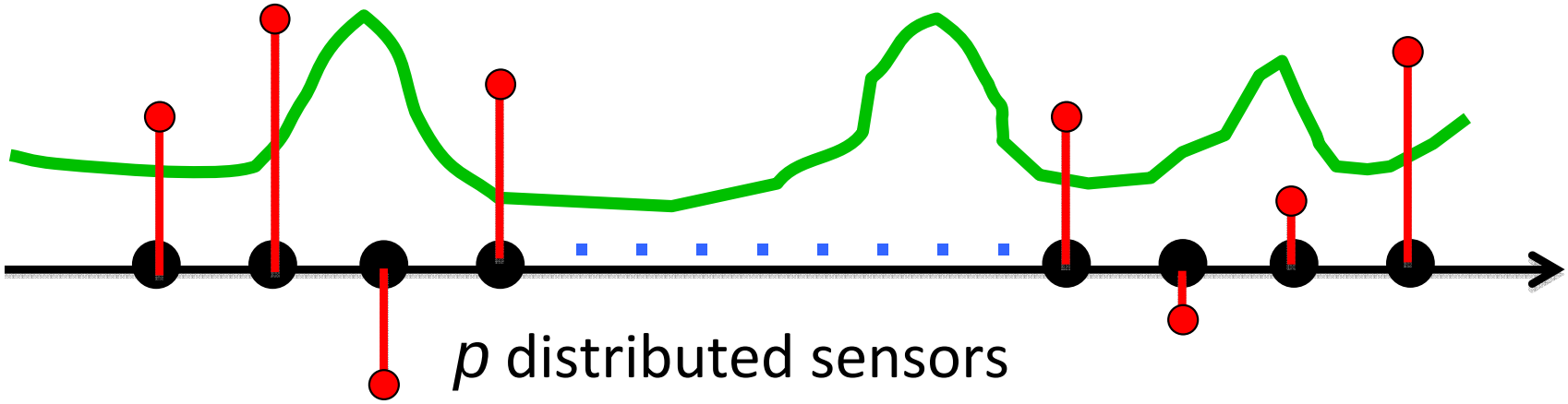


A sensor network application

Simple observation model using distributed sensors:

Class 1: $\mathbf{Y} = \mathbf{h} + \text{noise}$ Class 2: $\mathbf{Y} = -\mathbf{h} + \text{noise}$

Typical noisy observation given class 1

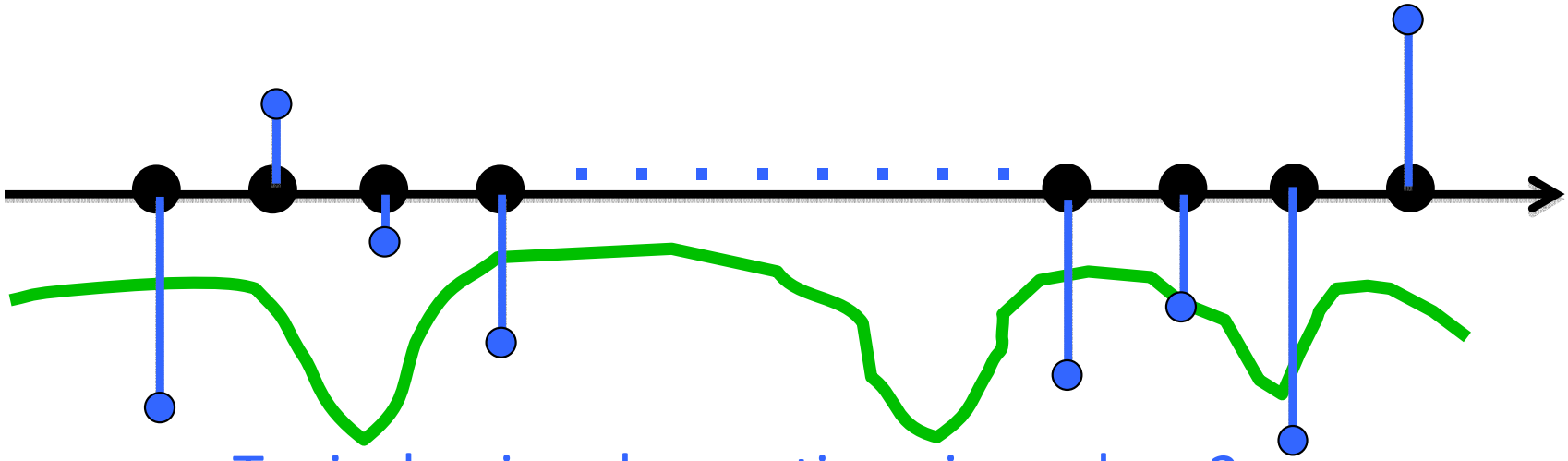


A sensor network application

Simple observation model using distributed sensors:

Class 1: **Y** = **h** + noise

Class 2: **Y** = -**h** + noise



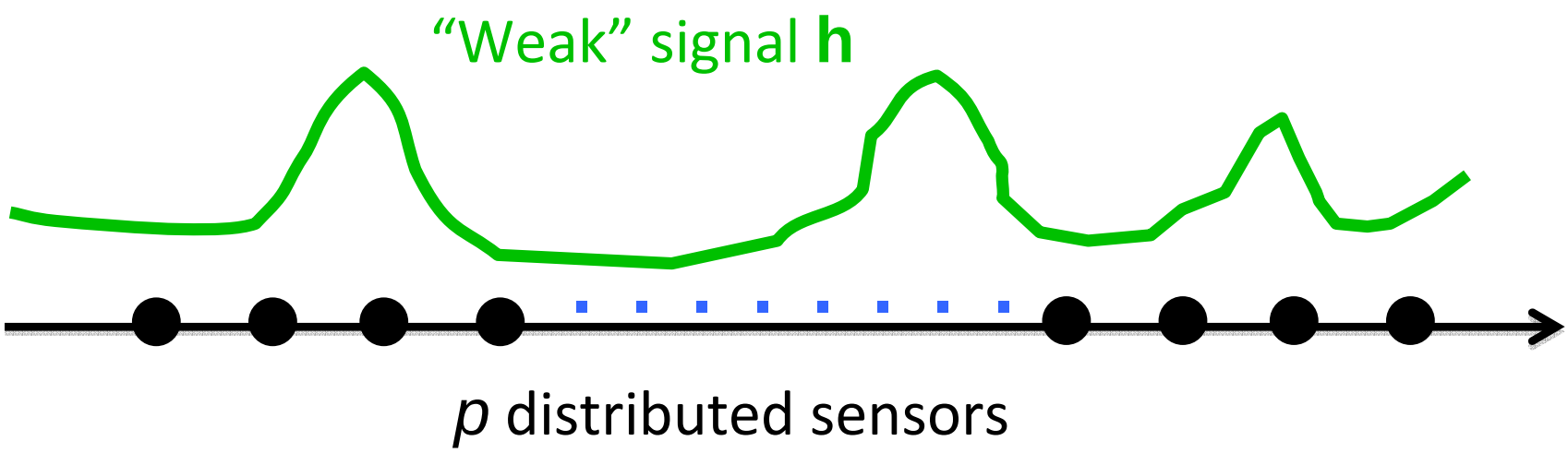
Typical noisy observation given class 2

A sensor network application

Simple observation model using distributed sensors:

Class 1: **Y** = **h** + noise

Class 2: **Y** = -**h** + noise



Given:

n noisy observations of the "weak" signal per sensor from each class

Decide:

If a new set of observations are coming from class 1 or class 2

Classifiers Considered

- Baseline:
  1. Full Bayes
- Unstructured:
  2. Full ML
- Structure-based:
  3. Full structure
  4. Structured ML
  5. Structured sparsity

Sensing model

$$\mathbf{Y}_i = \mathbf{h}X_i + \mathbf{Z}_i$$

$$\begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1} = \begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1} X_i + \begin{bmatrix} \\ \\ \end{bmatrix}_{p\times 1}$$

i = 1 (class 1)
i = 2 (class 2)

Likelihood

$$p(\mathbf{Y}|\mathbf{h}, class\ i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$$

1. Baseline: Full Bayes Classifier

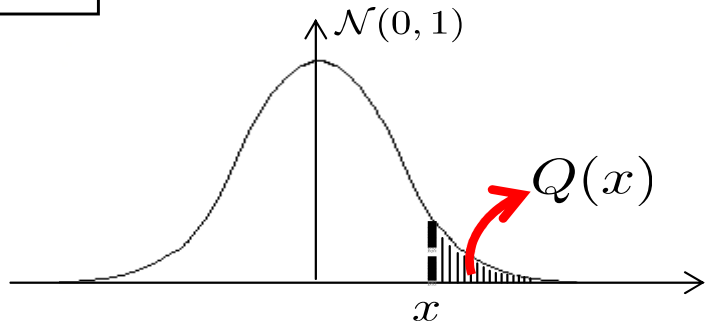
- Optimal test (Bayes rule): $\mathbf{1}\left\{\frac{p(\mathbf{Y}|\mathbf{h}, class\ 2)}{p(\mathbf{Y}|\mathbf{h}, class\ 1)} \geq 1\right\}$

- Classification rule: $\delta_{Bayes}(\mathbf{y}) = \mathbf{1}\left\{\boldsymbol{\Delta}^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) \geq 0\right\}$

$$\begin{aligned} \boldsymbol{\Delta} &= \mathbb{E}[\mathbf{Y}|class2] - \mathbb{E}[\mathbf{Y}|class1] &&= (m_2 - m_1)\mathbf{h} \\ \boldsymbol{\mu} &= \frac{1}{2}(\mathbb{E}[\mathbf{Y}|class2] + \mathbb{E}[\mathbf{Y}|class1]) &&= \frac{(m_2 + m_1)}{2}\mathbf{h} \\ \boldsymbol{\Sigma} &= \mathrm{cov}(\mathbf{Y}) &&= \sigma_x^2 \mathbf{h}\mathbf{h}^T + \sigma_z^2 \mathbf{I} \end{aligned}$$

- Misclassification probability:

$$P_M^{\delta_{Bayes}} = Q\left(\frac{m_2 - m_1}{2\sqrt{\sigma_x^2 + \frac{\sigma_z^2}{\|\mathbf{h}\|^2}}}\right)$$



Fixed difficulty level ➔
$m_1, m_2, \sigma_x, \sigma_z$ & $||\mathbf{h}||$ fixed!

BOSTON UNIVERSITY

2. Unstructured: Full ML

- Recall full Bayes classifier:

$$\delta_{Bayes}(\mathbf{y}) = \mathbf{1}\left\{\boldsymbol{\Delta}^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) \geq 0\right\}$$

- Unstructured full ML:
  - Use $\delta_{Bayes}(\mathbf{y})$ with ML estimates of $\hat{\boldsymbol{\Delta}}, \hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\mu}}$

- Empirical Fisher rule:

$$\delta_{Unstructured}(\mathbf{y}) = \mathbf{1}\left\{\hat{\boldsymbol{\Delta}}^T \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}) \geq 0\right\}$$

2. Unstructured: Full ML (continued)

Misclassification probability:

$$P_M^{\delta_{Unstructured}} = Q\left(\frac{\hat{\boldsymbol{\Delta}}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\Delta}}}{2(\hat{\boldsymbol{\Delta}}^T \hat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{\Sigma} \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\Delta}})^{1/2}} \right)$$

Theorem 1

If $n/p \to 0$ and $\| \mathbf{h} \| = \text{constant}$, then $P_M^{\delta_{Unstructured}} \xrightarrow{p} 1/2$

Ignoring structure and estimating all parameters is not a good strategy for inference in high-dimensional scenarios with few samples

3. Full Structure

- Recall sensing model:

$$\mathbf{Y}_i = \mathbf{h} X_i + \mathbf{Z}_i$$

- Structure-based Full Structure:
  - $\mathbf{h}$ KNOWN
  - Project data onto $\mathbf{h}$, then use ML estimates of parameters

- Projected Empirical Fisher rule:

$$\delta_{Full-Structure}(\mathbf{y}) = \mathbf{1}\left\{\mathbf{h}^T(\mathbf{y} - \hat{\boldsymbol{\mu}})\,\mathrm{sign}(\hat{\boldsymbol{\Delta}}^T\mathbf{h}) \geq 0\right\}$$

3. Full Structure (continued)

Misclassification probability:

$$P_M^{\delta_{Full-Structure}} = \frac{1}{2}\sum_{i=1}^{2} Q\left((-1)^{i-1} \frac{\mathbf{h}^T(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_i)\,\mathrm{sign}(\hat{\boldsymbol{\Delta}}^T\mathbf{h})}{(\mathbf{h}^T\boldsymbol{\Sigma}\mathbf{h})^{1/2}} \right)$$

Theorem 2

If $n \to \infty$ and $\| \mathbf{h} \| = \text{constant}$, then $P_M^{\delta_{Full-Structure}} \xrightarrow{p} P_M^{\delta_{Bayes}}$

Full knowledge of low-dimensional structure knowledge yields Bayes-optimal performance.

4. Structured ML

- Recall sensing model:

$$\mathbf{Y}_i = \mathbf{h} X_i + \mathbf{Z}_i$$

- Structured ML:
  - Sensing model known but $\mathbf{h}$ UNKNOWN
  - Use ML estimate of $\Delta = (m_2\text{-}m_1)\mathbf{h}$ as a proxy for $\mathbf{h}$
    - Recall: if $\mathbf{h}$ known, $P_{\text{error}} \rightarrow P_{\text{Bayes}}$

- Projected Empirical Fisher rule:

$$\delta_{Structured-ML}(\mathbf{y}) = \mathbf{1}\left\{ \hat{\boldsymbol{\Delta}}^T (\mathbf{y} - \hat{\boldsymbol{\mu}}) \geq 0 \right\}$$

⟸ Essentially, Naive Bayes

4. Structured ML (continued)

Misclassification probability:

$$P_M^{\delta_{Structured-ML}} = \frac{1}{2}\sum_{i=1}^{2} Q\left((-1)^{i-1} \frac{\hat{\boldsymbol{\Delta}}^T(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_i)}{(\hat{\boldsymbol{\Delta}}^T \boldsymbol{\Sigma} \hat{\boldsymbol{\Delta}})^{1/2}} \right)$$

Theorem 3

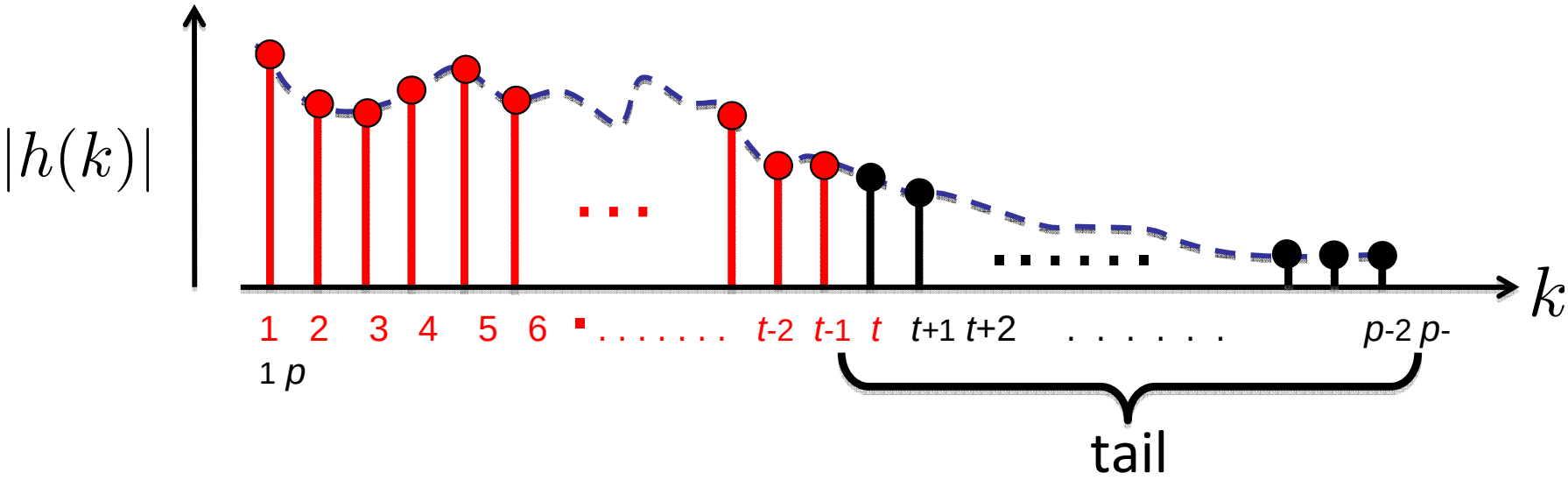
If $n/p \to 0$ and $\| \mathbf{h} \| = \text{constant}$, then $P_M^{\delta_{Structured-ML}} \xrightarrow{p} 1/2$

➔ Simple ML estimation of low-dimensional structure does not work.

➔ Additional structure is needed!

5. Structured Sparsity

- "Sparsity" assumption: energy in **h** localized in few components



- Tail energy goes to zero: As $t, p \to \infty$, $\sum_{k=t+1}^{p} h^2(k) \to 0$

5. Structured Sparsity (continued)

- Truncation estimator $\hat{\boldsymbol{\Delta}}_t(j) = \begin{cases} \hat{\boldsymbol{\Delta}}(j) & \text{if} \quad j \le t \\ 0 & \text{else} \end{cases}$

Theorem 4

(a) If $p, n, t \to \infty$ and $n/t \to \infty$, then $E[\| \hat{\boldsymbol{\Delta}}_t - \boldsymbol{\Delta} \|^2] \to 0$

(b) If $n/p \to 0$ and $\| \mathbf{h} \| =$ constant, then $P_M^{\delta_{Structured-Sparsity}} \xrightarrow{p} P_M^{\delta_{Bayes}}$

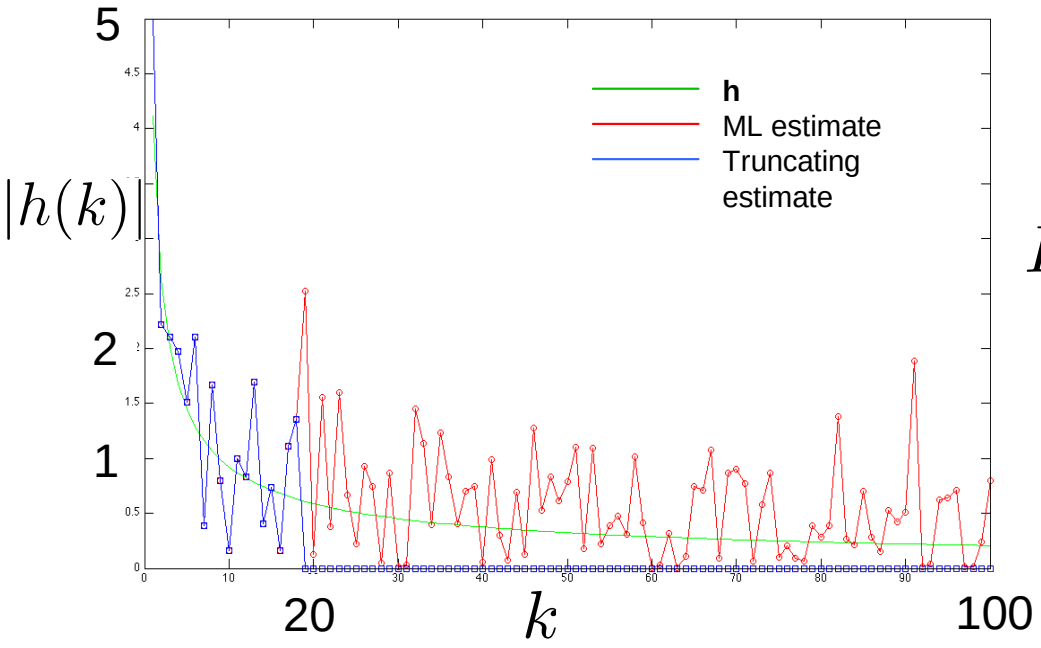
- Knowledge of sensing structure + sparsity ➔

Asymptotically optimum performance!

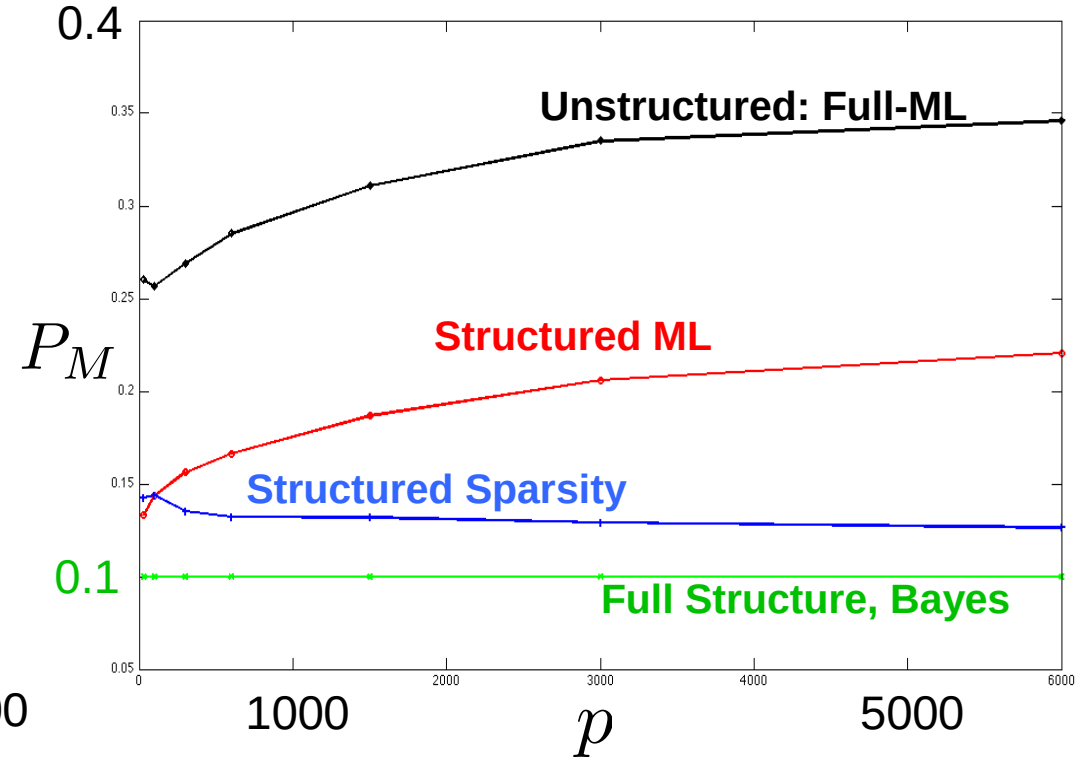
Simulation – Polynomially decaying $\mathbf{h}$

$$P_M^{\delta_{Bayes}} = 0.1,\ \|\mathbf{h}\| = 2,\ \sigma_x = 1,\ \sigma_z = 2,\ n = p^{0.7},\ t = n^{0.95}$$

Polynomially decaying $\mathbf{h}$ and its ML & truncating estimates



Misclassification Probability



Concluding remarks

- In many problems of engineering interest
  - *# samples << data-dimension*
  - there exists a latent low-dimensional sensing structure

- Totally ignoring sensing structure ➔ no better than guessing

- General knowledge of sensing structure alone, however, is insufficient

- Knowledge of sensing structure
  +
  Additional structure in **h**

Optimum asymptotic classification performance