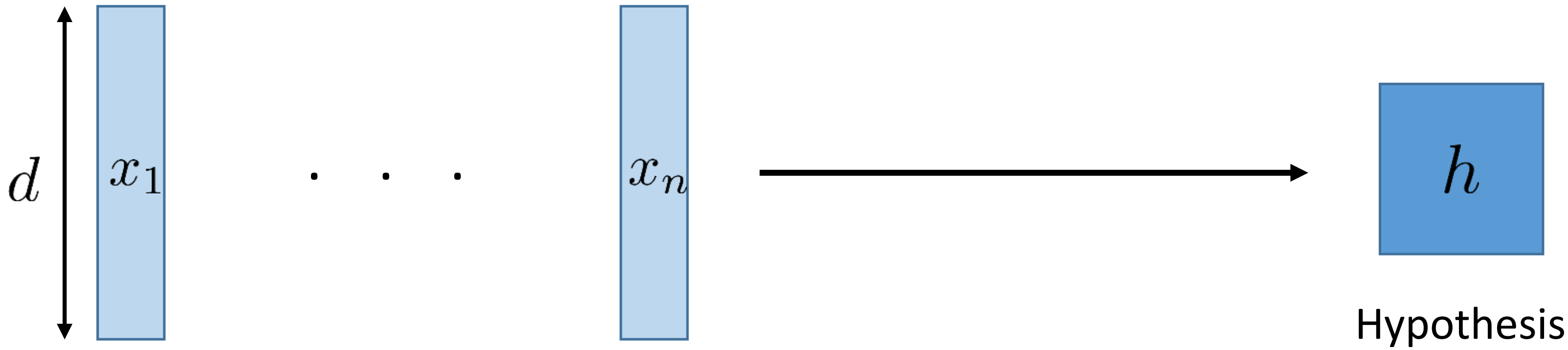
Random Moments for Sketched Mixture Learning

Nicolas Keriven[12], Rémi Gribonval[2],
Gilles Blanchard[3], Yann Traonmilin[2]
[1]Université Rennes 1
[2]Inria Rennes Bretagne-atlantique
[3]University of Potsdam

SPARS 2017

Outline

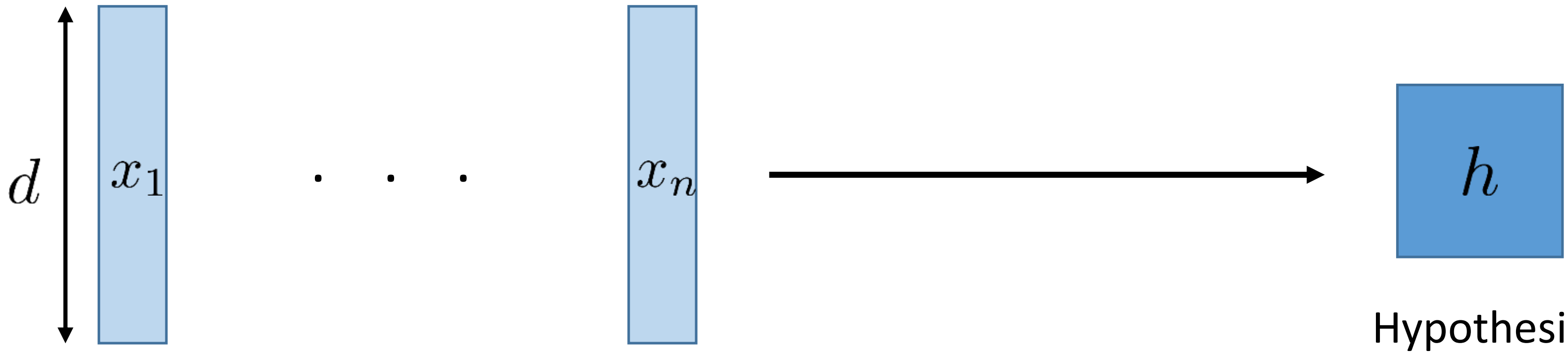
Statistical Learning



Database of vectors i.i.d. π_0

Hypothesis

Statistical Learning

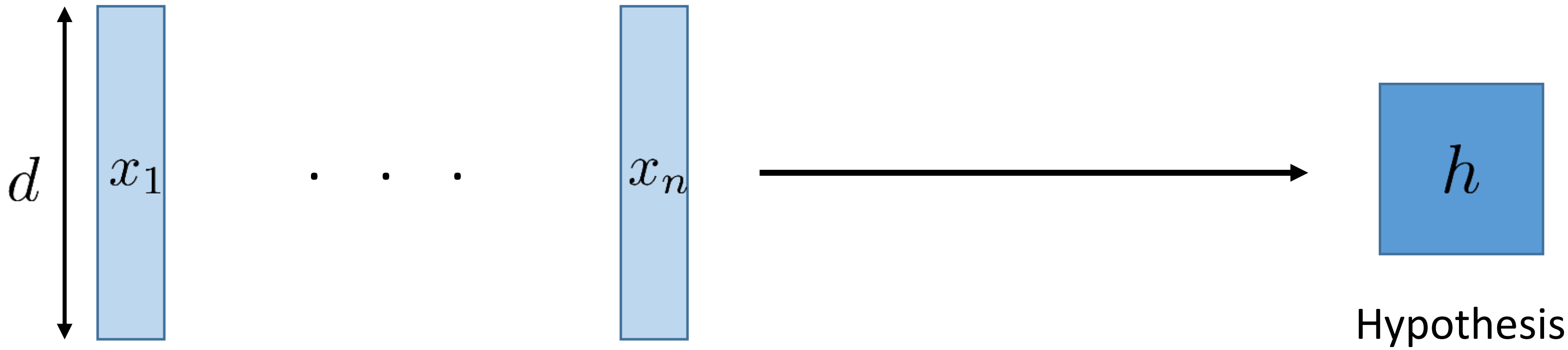


Hypothesis

Database of vectors i.i.d. π_0

- PCA : $\mathbf{x} \in Span(h_1, ..., h_k)$
- Classification : $< h, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = h(\mathbf{x})$
- **k-means** : $h = \{c_l\}_{l=1}^k$
- **Density estimation** : $\mathbf{x} \sim \pi_h$

Statistical Learning

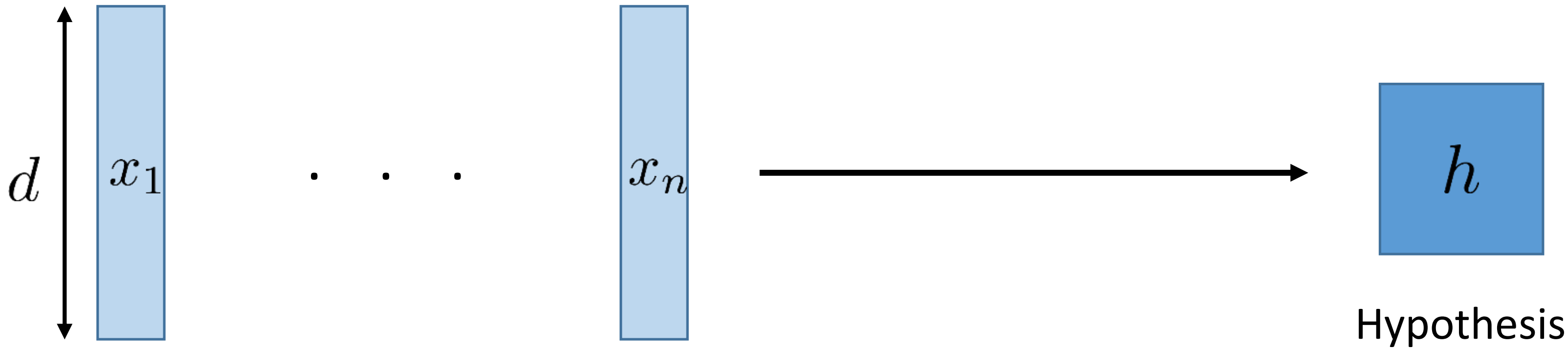


Database of vectors i.i.d. π_0

Loss function $\ell(x, h) \in \mathbb{R}$

- PCA : $\mathbf{x} \in Span(h_1, ..., h_k)$
- Classification : $< h, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = h(\mathbf{x})$
- **k-means** : $h = \{c_l\}_{l=1}^k$
- **Density estimation** : $\mathbf{x} \sim \pi_h$

Statistical Learning



Hypothesis

Database of vectors i.i.d. π_0

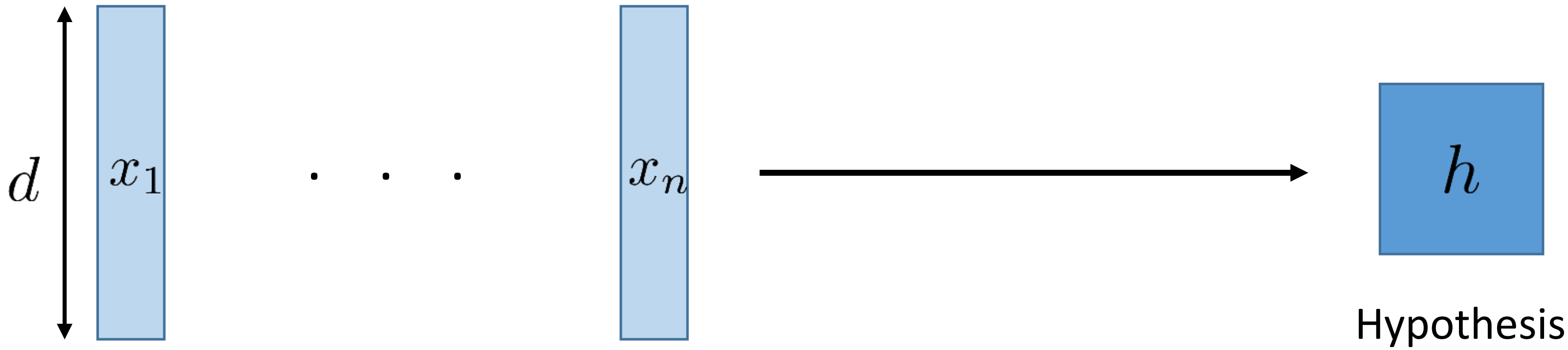
Loss function $\ell(x,h) \in \mathbb{R}$

Goal : Minimize Expected Risk

$$h^\star = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{x \sim \pi_0} \ell(x,h)$$

- PCA : $\mathbf{x} \in Span(h_1, ..., h_k)$
- Classification : $< h, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = h(\mathbf{x})$
- **k-means** : $h = \{c_l\}_{l=1}^k$
- **Density estimation** : $\mathbf{x} \sim \pi_h$

Statistical Learning



Database of vectors i.i.d. π_0

Loss function $\ell(x, h) \in \mathbb{R}$

Goal : Minimize Expected Risk

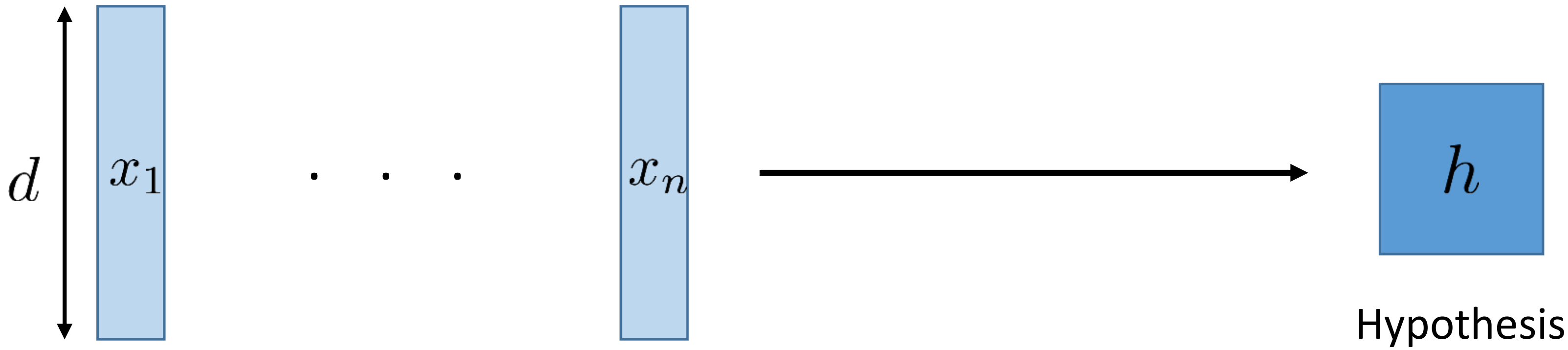
$$h^\star = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

Empirical Risk Minimization (ERM)

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, h)$$

- PCA : $\mathbf{x} \in Span(h_1, ..., h_k)$
- Classification : $< h, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = h(\mathbf{x})$
- **k-means** : $h = \{c_l\}_{l=1}^k$
- **Density estimation** : $\mathbf{x} \sim \pi_h$

Statistical Learning



Database of vectors i.i.d. π_0

Loss function $\ell(x, h) \in \mathbb{R}$

Goal : Minimize Expected Risk

$$h^\star = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

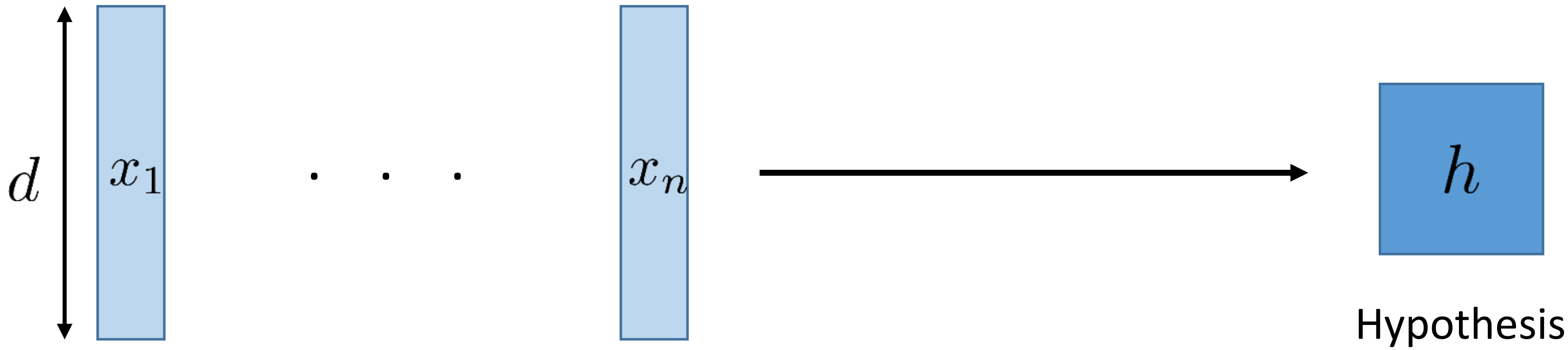
Empirical Risk Minimization (ERM)

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, h)$$

- PCA : $\mathbf{x} \in Span(h_1, ..., h_k)$
- Classification : $< h, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = h(\mathbf{x})$
- **k-means** : $h = \{c_l\}_{l=1}^k$
- **Density estimation** : $\mathbf{x} \sim \pi_h$

Large d or n

Statistical Learning



Database of vectors i.i.d. π_0

Loss function $\ell(x, h) \in \mathbb{R}$

Goal : Minimize Expected Risk

$$h^\star = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

Empirical Risk Minimization (ERM)

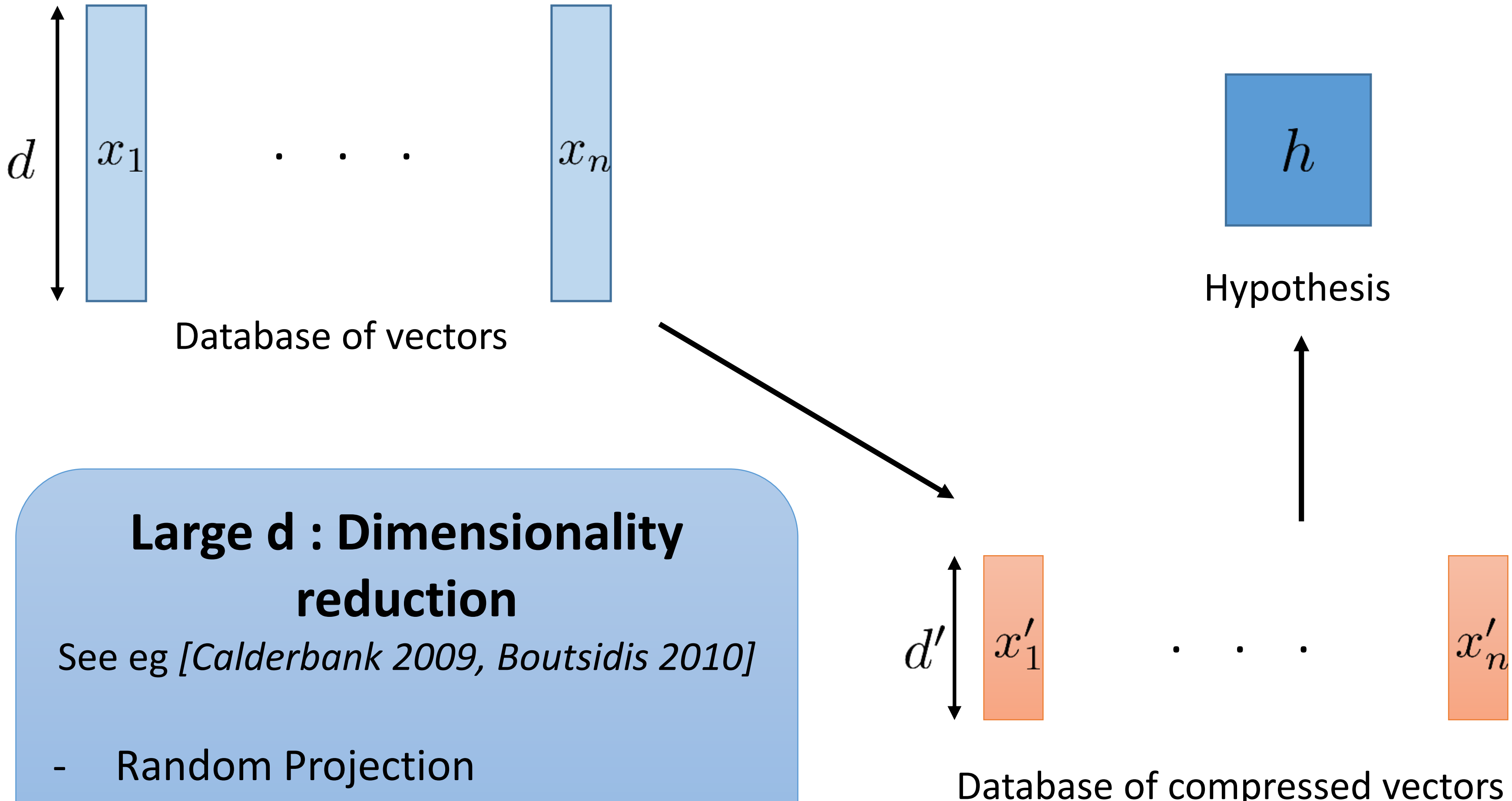
$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, h)$$

- PCA : $\mathbf{x} \in Span(h_1, ..., h_k)$
- Classification : $< h, \Phi(\mathbf{x}) >$
- Regression : $\mathbf{y} = h(\mathbf{x})$
- **k-means** : $h = \{c_l\}_{l=1}^k$
- **Density estimation** : $\mathbf{x} \sim \pi_h$

Large d or n

Compress the database before learning

Compressive Statistical Learning

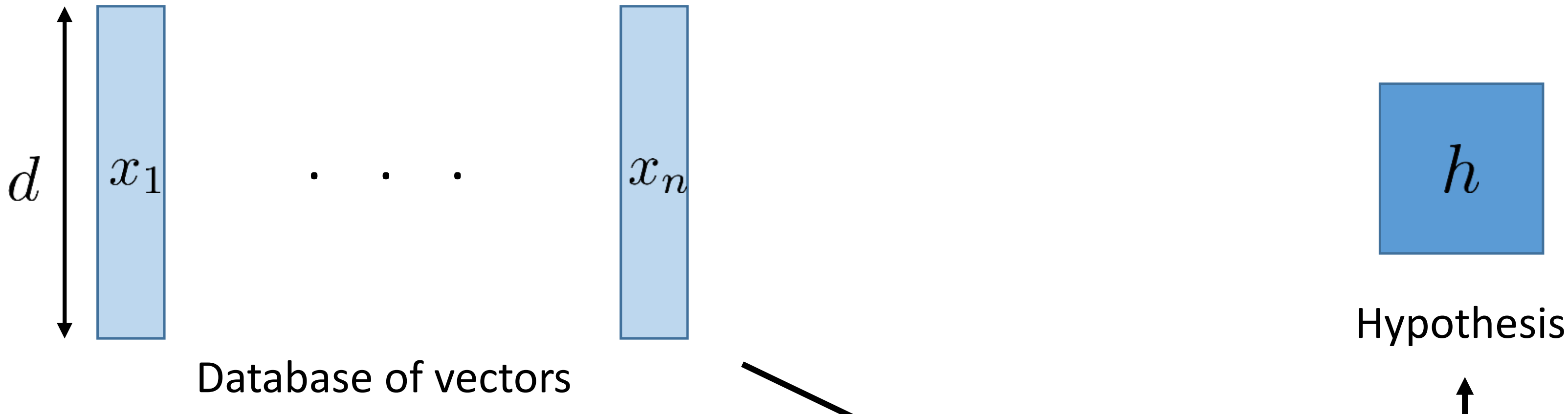


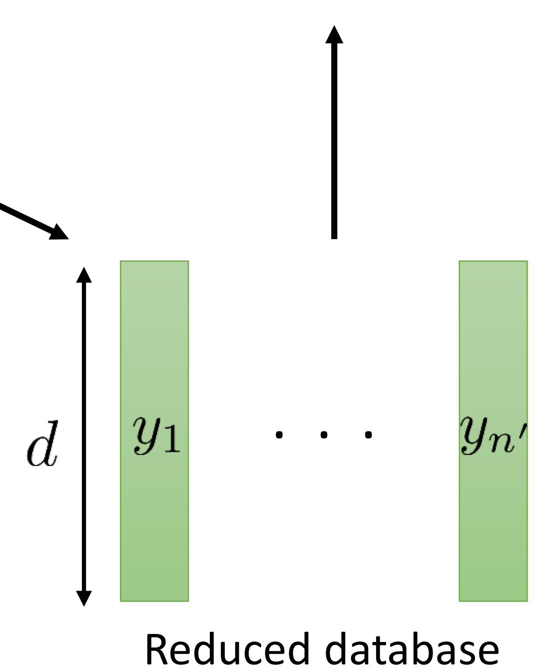
Large d : Dimensionality reduction

See eg *[Calderbank 2009, Boutsidis 2010]*

- Random Projection
- Feature selection

Compressive Statistical Learning



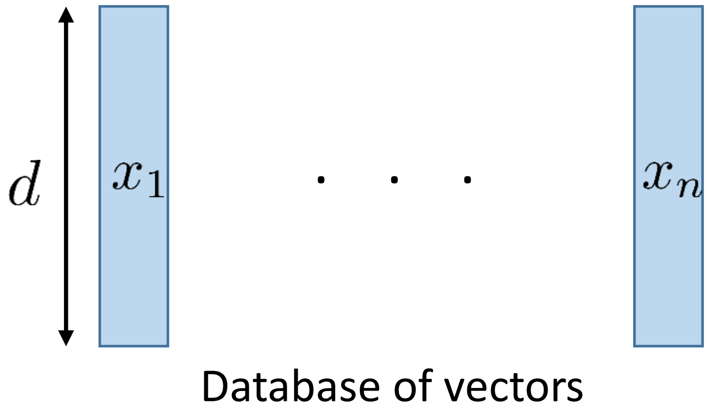


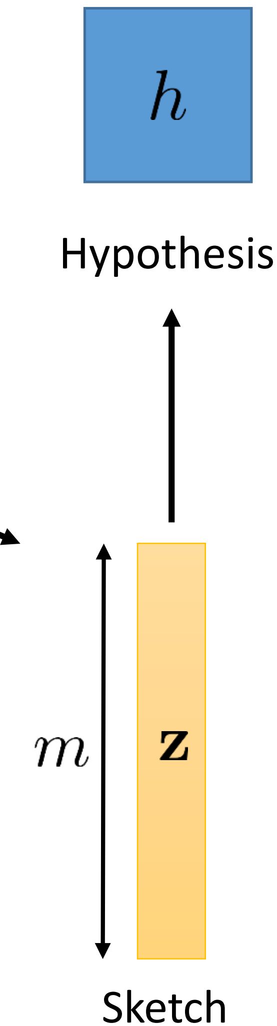
Large n : Subsampling, coresets

See eg *[Feldman 2010]*

- Uniform sampling (naive)
- Adaptive, weighted sampling
- Hierarchical construction

Compressive Statistical Learning





Linear sketch

See *[Thaper 2002, Cormode 2011]*

- Sketch of union of databases = sum of sketches
- **Extremely convenient for streaming / parallel computing**
- **Used for simple queries. Can we do learning ?**

Random Sketching operator

Linear sketch = Empirical generalized moments…

$$\mathbf{z} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} \Phi(x_i)$$

Random Sketching operator

Linear sketch = Empirical generalized moments…

$$\mathbf{z} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} \Phi(x_i)$$

… i.e. *linear measurements of underlying probability distribution*

$$\mathbf{z} \approx \mathbb{E}_{x \sim \pi_0} \Phi(x) = \mathcal{A}\pi_0$$

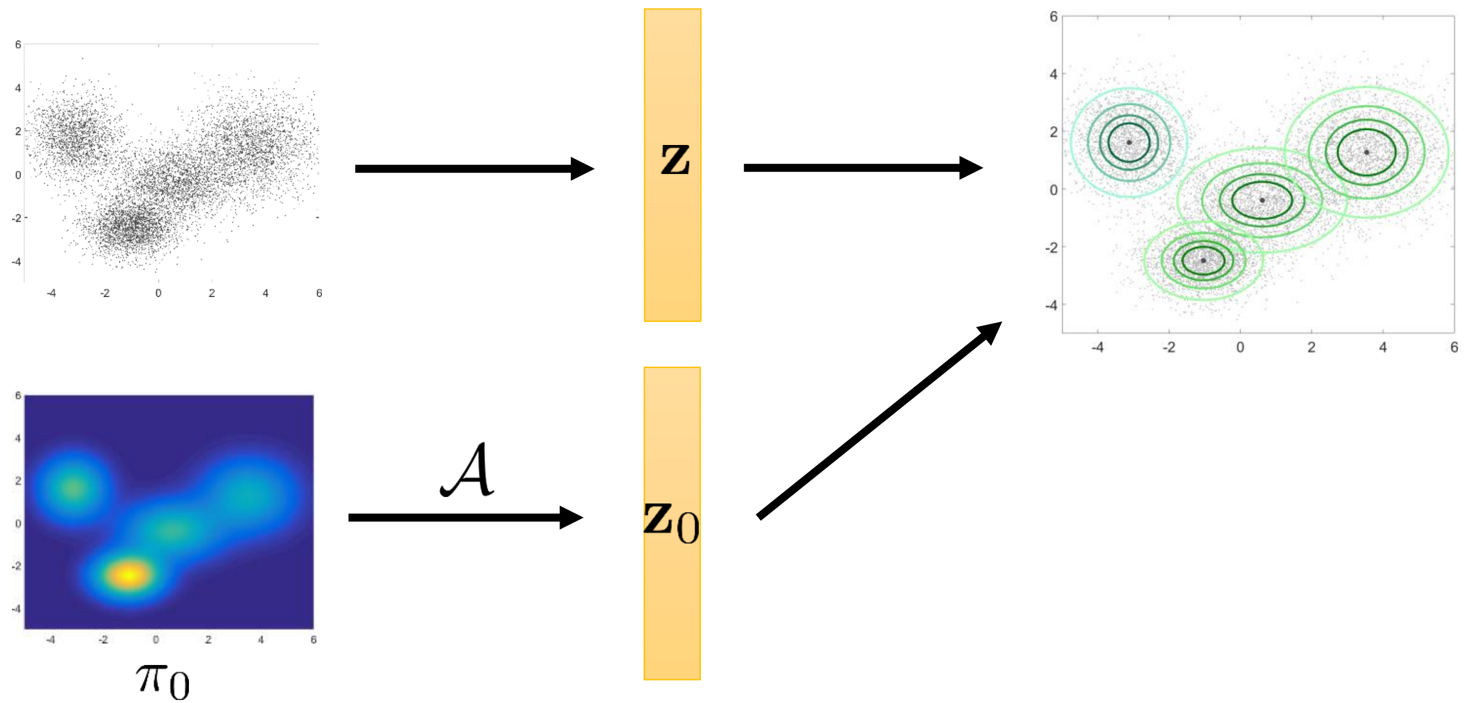
Random Sketching operator

Linear sketch = Empirical generalized moments…

$$\mathbf{z} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} \Phi(x_i)$$

… i.e. *linear measurements of underlying probability distribution*

$$\mathbf{z} \approx \mathbb{E}_{x \sim \pi_0} \Phi(x) = \mathcal{A}\pi_0$$



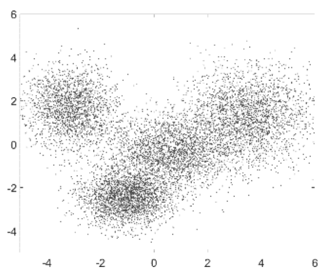
Random Sketching operator

Linear sketch = Empirical generalized moments…

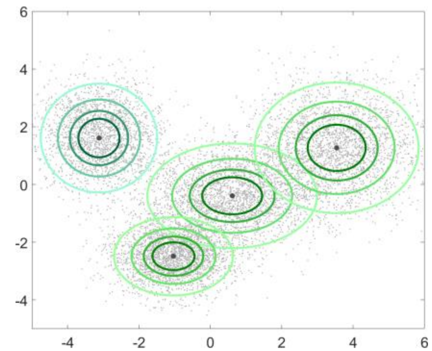
$$\mathbf{z} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} \Phi(x_i)$$

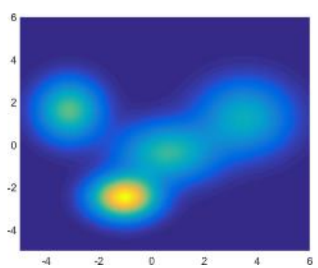
… i.e. *linear measurements of underlying probability distribution*

$$\mathbf{z} \approx \mathbb{E}_{x\sim\pi_0} \Phi(x) = \mathcal{A}\pi_0$$



$\mathbf{z}$





π_0

$\mathcal{A}$

$\mathbf{z}_0$

Reminiscent of Compressive Sensing : Random design of $\mathcal{A}$

Outline

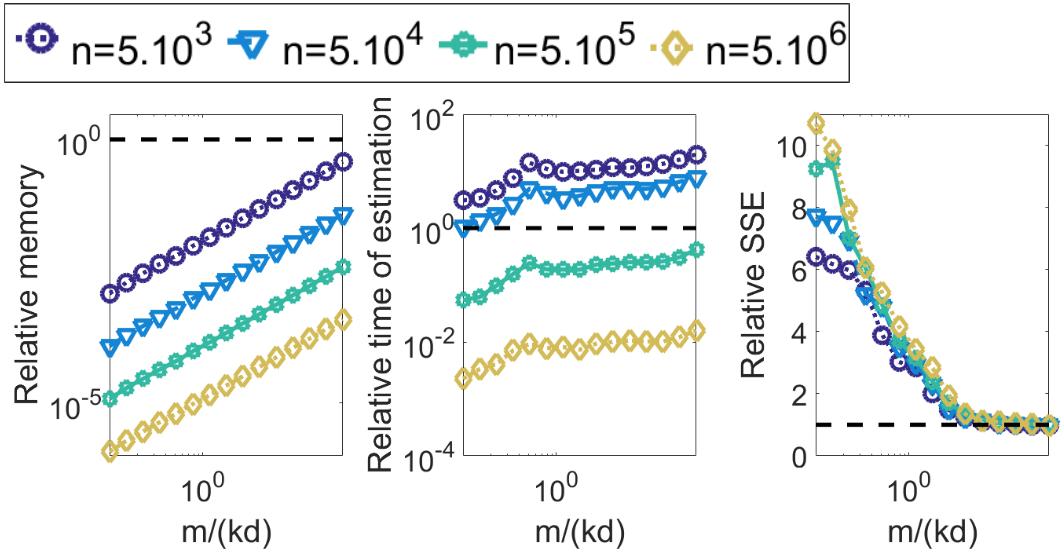
Experimental illustration

Compressive Learning-OMP algorithm *[Keriven 2015,2016]*
(OMP + non-convex updates)

Experimental illustration

k-means (d=10, k=10)

Compressive Learning-OMP algorithm *[Keriven 2015,2016]*
(OMP + non-convex updates)



Comparison with

- *Matlab's* `kmeans`
- *VLFeat's* `gmm`

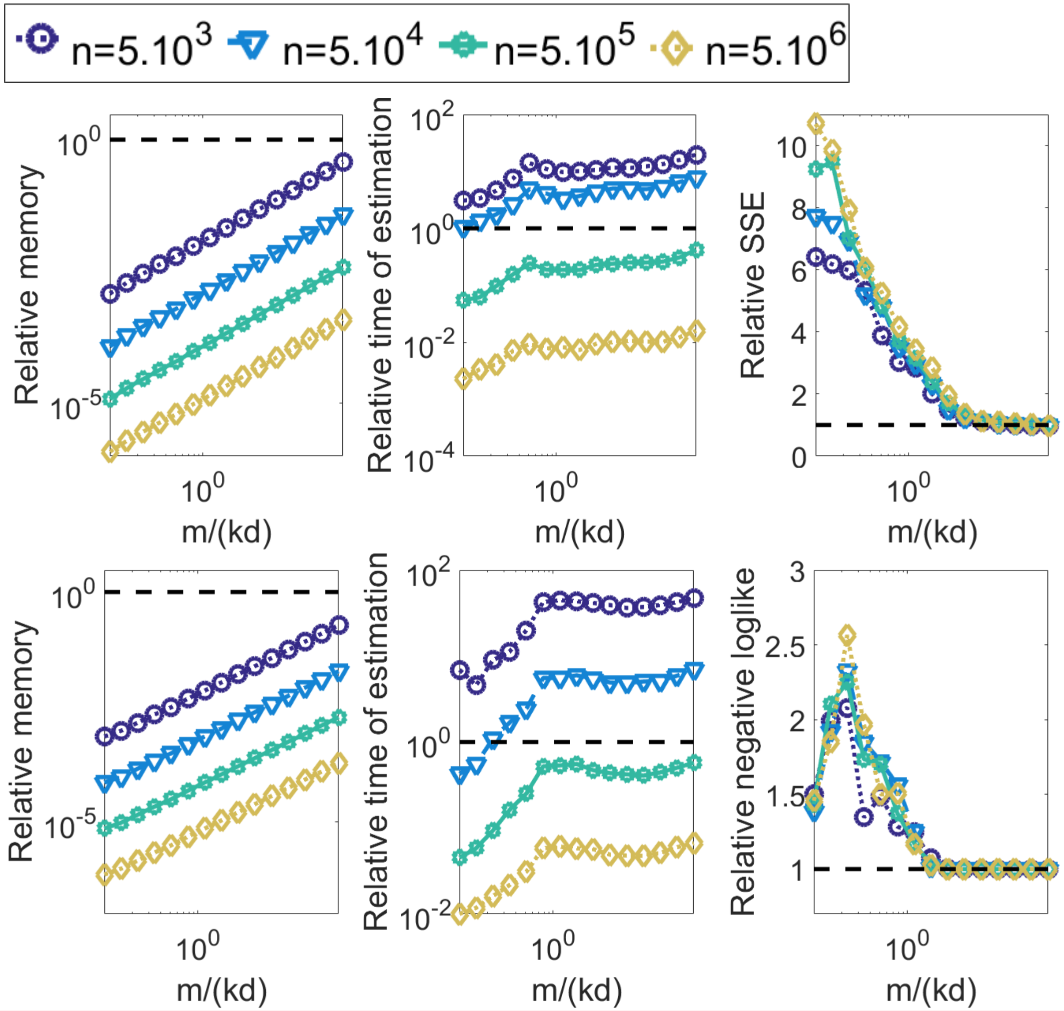
- Faster and more memory efficient on large databases
- Number of measurements does not depend on n

Experimental illustration

k-means (d=10, k=10)

Compressive Learning-OMP algorithm *[Keriven 2015,2016]*

(OMP + non-convex updates)



Comparison with

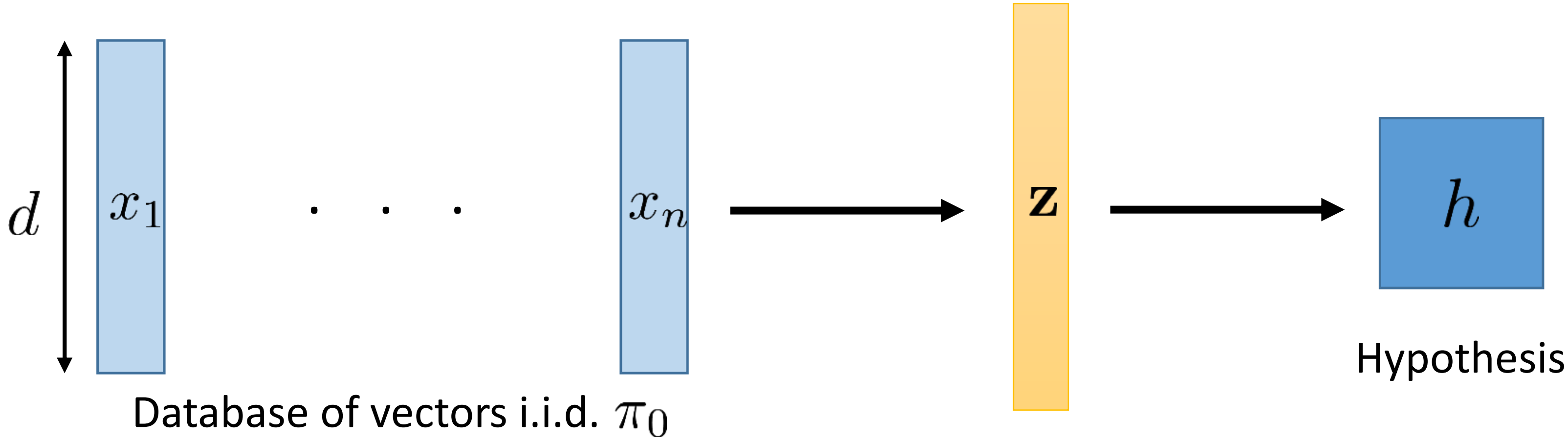
- *Matlab's* `kmeans`
- *VLFeat's* `gmm`

- Faster and more memory efficient on large databases
- Number of measurements does not depend on n

GMMs (d=10, k=10)

Outline

Statistical Learning



Loss function $\ell(x, h)$

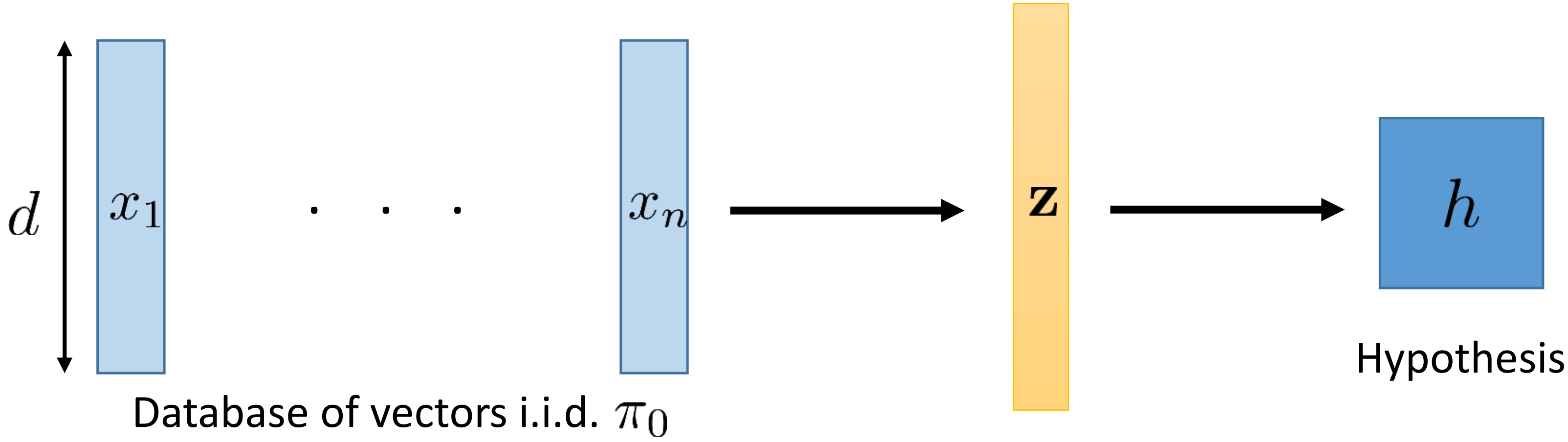
$$\mathbf{z} = \frac{1}{n}\sum_{i=1}^{n} \Phi(x_i)$$

$$\mathcal{A}\pi = \mathbb{E}_{x\sim\pi}\Phi(x)$$

Goal : Minimize Expected Risk

$$h^\star = \arg\min_{h\in\mathcal{H}} \mathbb{E}_{x\sim\pi_0}\ell(x, h)$$

Statistical Learning



Loss function $\ell(x, h)$

Goal : Minimize Expected Risk

$$h^\star = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

$$\mathbf{z} = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i)$$

$$\mathcal{A}\pi = \mathbb{E}_{x \sim \pi} \Phi(x)$$

Here:

- k-means
- GMM with known covariance

k-means

Hyp. class $h = \{c_1, ..., c_k\} \subset \mathbb{R}^d$

k-means

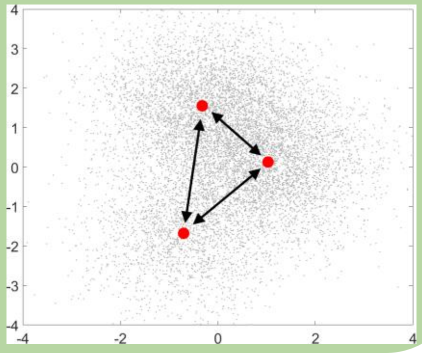
Hyp. class $h = \{c_1, ..., c_k\} \subset \mathbb{R}^d$

$\mathcal{H}_{k,\varepsilon,M}$

- M - bounded domain

(centroids, not samples)

- ε - separation



k-means

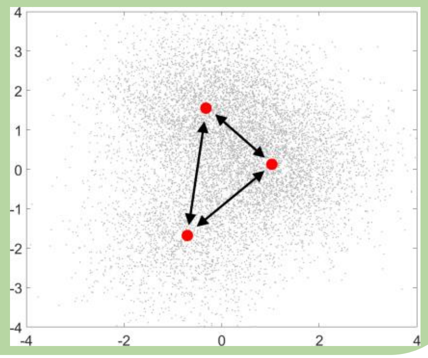
Hyp. class $h = \{c_1, ..., c_k\} \subset \mathbb{R}^d$

$\mathcal{H}_{k,\varepsilon,M}$

- M - bounded domain

(centroids, not samples)

- ε - separation



Loss function $\ell(x,h) = \min_{1 \leq l \leq k} \|x - c_l\|_2^2$

k-means

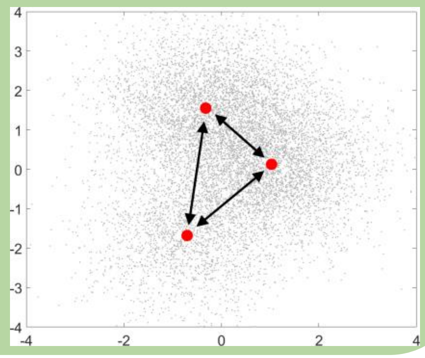
Hyp. class $h = \{c_1, ..., c_k\} \subset \mathbb{R}^d$

$\mathcal{H}_{k,\varepsilon,M}$

- M - bounded domain

(centroids, not samples)

- ε - separation



Loss function $\ell(x, h) = \min_{1 \leq l \leq k} \|x - c_l\|_2^2$

Sketching operator

$\{\omega_1, ..., \omega_m\} \subset (\mathbb{R}^d)^m$

$$\Phi(x) = \left[e^{-i\omega_j^T x} / c_{\omega_j} \right]_{j=1}^{m}$$

- (weighted) Random Fourier sampling
- « Smoothing » weights $c_\omega \propto \|\omega\|_2^2$

k-means

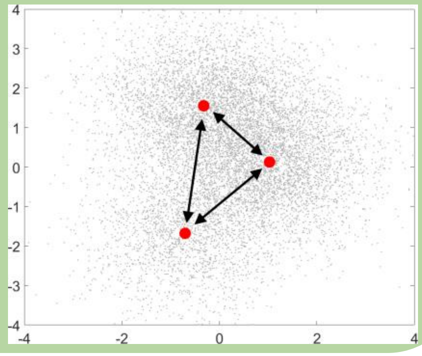
Hyp. class $h = \{c_1, ..., c_k\} \subset \mathbb{R}^d$

$\mathcal{H}_{k,\varepsilon,M}$

- M - bounded domain

(centroids, not samples)

- ε - separation



Loss function $\ell(x, h) = \min_{1 \leq l \leq k} \|x - c_l\|_2^2$

Sketching operator

$\{\omega_1, ..., \omega_m\} \subset (\mathbb{R}^d)^m$

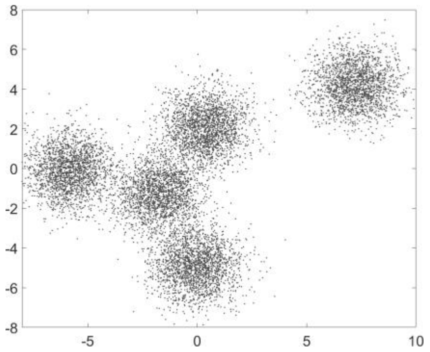
$$\Phi(x) = \left[e^{-i\omega_j^T x} / c_{\omega_j} \right]_{j=1}^m$$

- (weighted) Random Fourier sampling
- « Smoothing » weights $c_\omega \propto \|\omega\|_2^2$

$$\omega_j \overset{i.i.d.}{\sim} \Lambda(\omega) \propto c_\omega^2 \mathcal{N}(0, \sigma^2 \mathbf{Id})$$

$$\sigma^2 \propto \varepsilon^{-1}$$

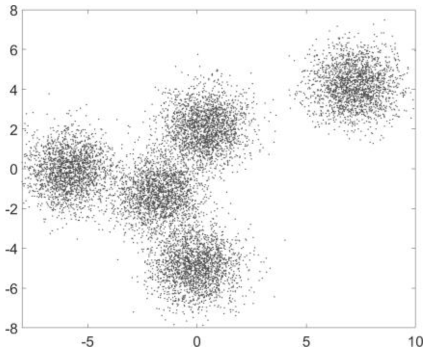
k-means: result

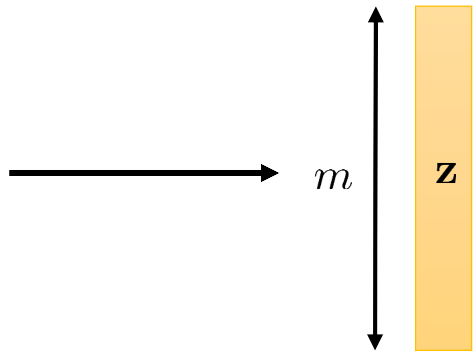


$$x_1, \dots, x_n \overset{i.i.d.}{\sim} \pi_0$$

$$R_{\pi_0}(h) = \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

k-means: result





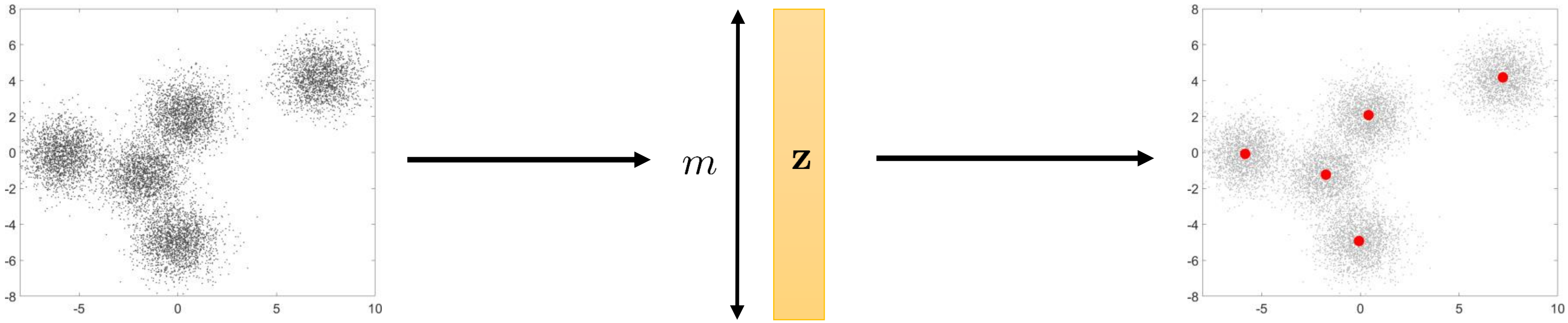
$$x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi_0$$

$$R_{\pi_0}(h) = \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

$$\omega_1, \ldots, \omega_m \overset{i.i.d.}{\sim} \Lambda$$

$\mathbf{z}$: (weighted) Random Fourier sampling

k-means: result



$x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi_0$

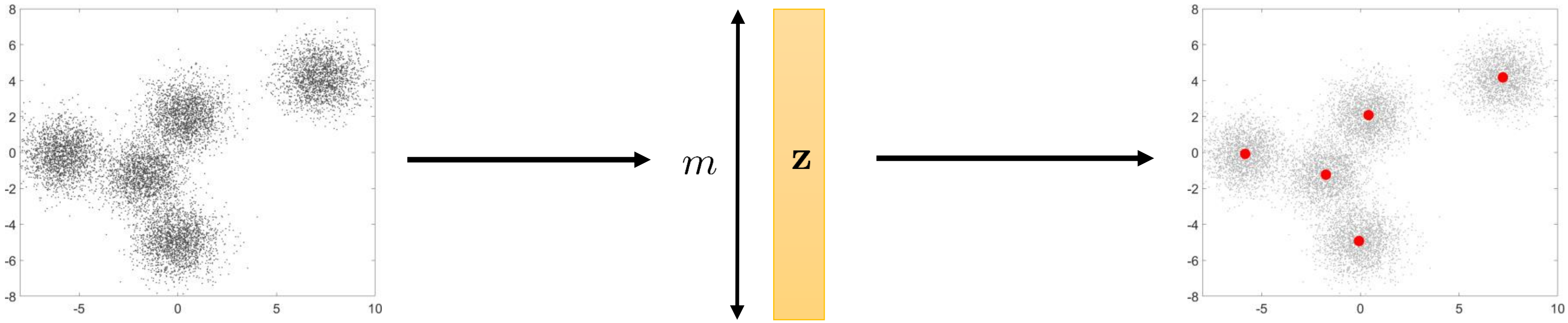
$\omega_1, \ldots, \omega_m \overset{i.i.d.}{\sim} \Lambda$

$R_{\pi_0}(h) = \mathbb{E}_{x \sim \pi_0} \ell(x, h)$

$\mathbf{z}$: (weighted) Random Fourier sampling

$$\hat{h} = \min_{h \in \mathcal{H}_{k,\varepsilon,M}} \min_{\alpha \geq 0, \sum_l \alpha_l = 1} \|\mathbf{z} - \mathcal{A}(\textstyle\sum_{l=1}^k \alpha_l \delta_{c_l})\|_2$$

k-means: result



$$x_1, \dots, x_n \overset{i.i.d.}{\sim} \pi_0 \qquad \omega_1, \dots, \omega_m \overset{i.i.d.}{\sim} \Lambda$$

$$R_{\pi_0}(h) = \mathbb{E}_{x\sim\pi_0} \ell(x, h)$$

$\mathbf{z}$: (weighted) Random Fourier sampling

$$\hat{h} = \min_{h\in\mathcal{H}_{k,\varepsilon,M}} \min_{\alpha\geq 0, \sum_l \alpha_l = 1} \|\mathbf{z} - \mathcal{A}(\textstyle\sum_{l=1}^k \alpha_l \delta_{c_l})\|_2$$

$$h^\star = \min_{h\in\mathcal{H}_{k,\varepsilon,M}} R_{\pi_0}(h)$$

If $m \geq \mathcal{O}\left(k^2 d^2 (\texttt{polylog}(k,d) + \log(M/\varepsilon))\right)$

w.h.p. on $x_i,\ \omega_j$ $\quad R_{\pi_0}(\hat{h}) \lesssim R_{\pi_0}(h^\star) + \mathcal{O}\left(\sqrt{1/n}\right)$

GMM with known covariance Σ

Hyp. class $h = \{(\mu_1, \alpha_1), ..., (\mu_k, \alpha_k)\} \subset \mathbb{R}^d \times \mathbb{R}_+$

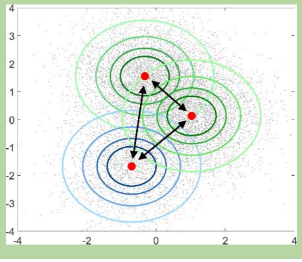
GMM with known covariance $\sum$

Hyp. class $h = \{(\mu_1, \alpha_1), ..., (\mu_k, \alpha_k)\} \subset \mathbb{R}^d \times \mathbb{R}_+$

$\mathcal{H}_{k,\varepsilon,M}$

- M - bounded domain
- $\varepsilon \geq \varepsilon_0$ separation

(means, not samples)



GMM with known covariance Σ

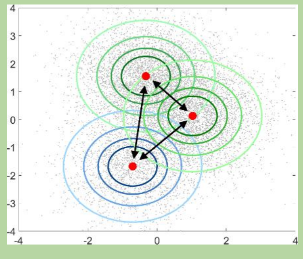
Hyp. class $h = \{(\mu_1, \alpha_1), ..., (\mu_k, \alpha_k)\} \subset \mathbb{R}^d \times \mathbb{R}_+$

$\mathcal{H}_{k,\varepsilon,M}$

- M - bounded domain

(means, not samples)

- $\varepsilon \geq \varepsilon_0$ separation



Loss function $\pi_h = \sum_{l=1}^{k} \alpha_l \mathcal{N}(\mu_l, \Sigma)$

$$\ell(x, h) = -\log \pi_h(x)$$

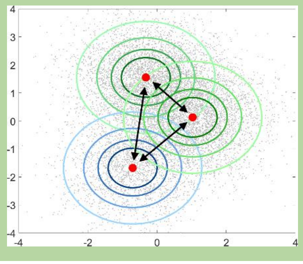
GMM with known covariance Σ

Hyp. class $h = \{(\mu_1, \alpha_1), ..., (\mu_k, \alpha_k)\} \subset \mathbb{R}^d \times \mathbb{R}_+$

$\mathcal{H}_{k,\varepsilon,M}$

- $\varepsilon \geq \varepsilon_0$ separation
- M - bounded domain

(means, not samples)



Loss function $\pi_h = \sum_{l=1}^k \alpha_l \mathcal{N}(\mu_l, \Sigma)$ $\ell(x, h) = -\log \pi_h(x)$

Sketching operator

- Random Fourier sampling

$\{\omega_1, ..., \omega_m\} \subset (\mathbb{R}^d)^m$

$$\Phi(x) = \left[e^{-i\omega_j^T x}\right]_{j=1}^m$$

GMM with known covariance Σ

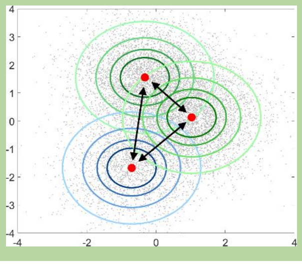
Hyp. class $h = \{(\mu_1, \alpha_1), ..., (\mu_k, \alpha_k)\} \subset \mathbb{R}^d \times \mathbb{R}_+$

$\mathcal{H}_{k,\varepsilon,M}$

- M - bounded domain

(means, not samples)

- $\varepsilon \geq \varepsilon_0$ separation



Loss function $\pi_h = \sum_{l=1}^k \alpha_l \mathcal{N}(\mu_l, \Sigma)$

$$\ell(x, h) = -\log \pi_h(x)$$

Sketching operator

$\{\omega_1, ..., \omega_m\} \subset (\mathbb{R}^d)^m$

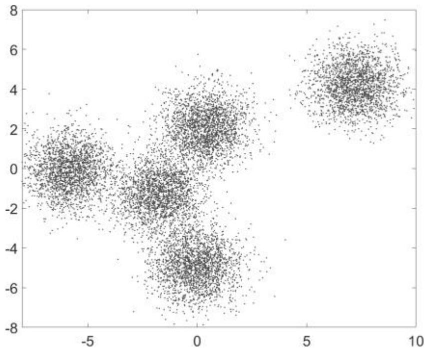
$$\Phi(x) = \left[e^{-i\omega_j^T x}\right]_{j=1}^m$$

- Random Fourier sampling

σ^2 linked to separation

$$\omega_j \overset{i.i.d.}{\sim} \Lambda(\omega) = \mathcal{N}(0, \sigma^2 \Sigma^{-1})$$

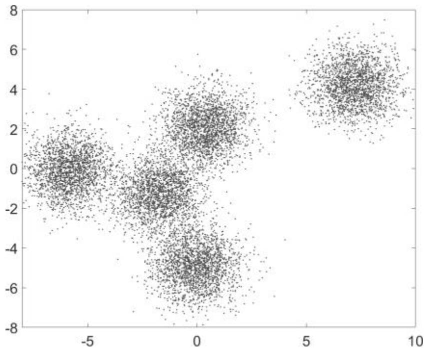
GMM: result

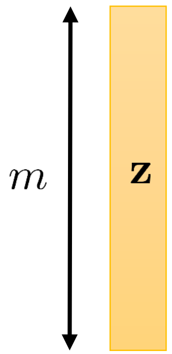


$$x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi_0$$

$$R_{\pi_0}(h) = \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

GMM: result





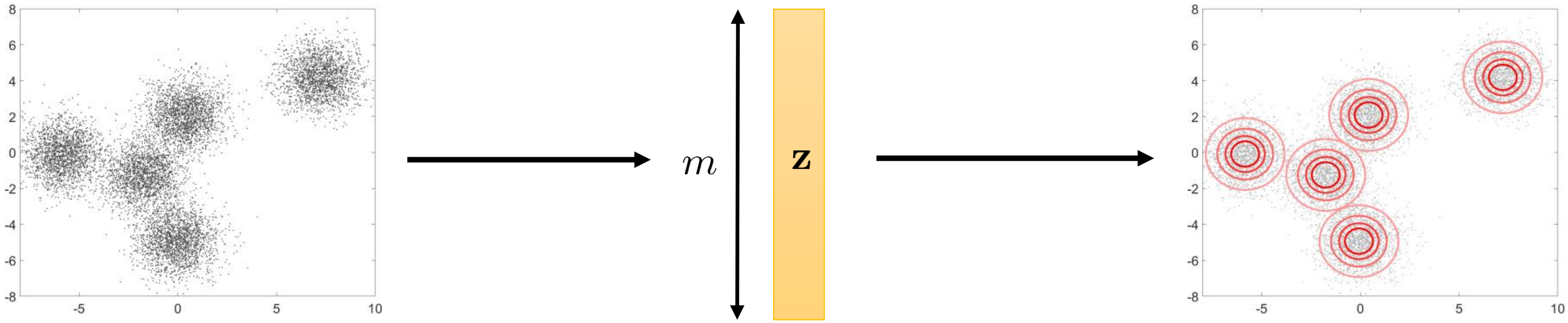
$$x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi_0$$

$$R_{\pi_0}(h) = \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

$$\omega_1, \ldots, \omega_m \overset{i.i.d.}{\sim} \Lambda$$

$\mathbf{z}$: Random Fourier sampling

GMM: result



$x_1, \ldots, x_n \overset{i.i.d.}{\sim} \pi_0$

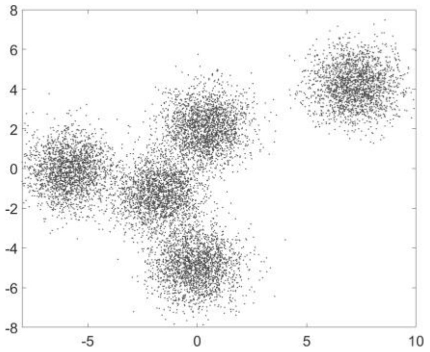
$\omega_1, \ldots, \omega_m \overset{i.i.d.}{\sim} \Lambda$

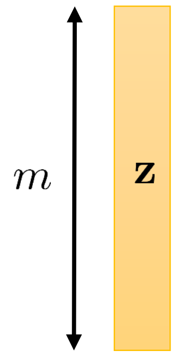
$R_{\pi_0}(h) = \mathbb{E}_{x \sim \pi_0} \ell(x, h)$

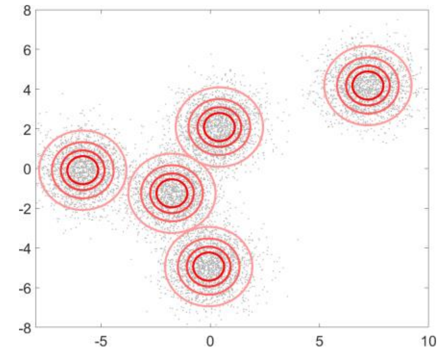
$\mathbf{z}$: Random Fourier sampling

$$\hat{h} = \min_{h \in \mathcal{H}_{k,\varepsilon,M}} \|\mathbf{z} - \mathcal{A}\pi_h\|_2^2$$

GMM: result







$$x_1, \dots, x_n \overset{i.i.d.}{\sim} \pi_0$$

$$R_{\pi_0}(h) = \mathbb{E}_{x \sim \pi_0} \ell(x, h)$$

$$\omega_1, \dots, \omega_m \overset{i.i.d.}{\sim} \Lambda$$

$\mathbf{z}$: Random Fourier sampling

$$\hat{h} = \min_{h \in \mathcal{H}_{k,\varepsilon,M}} \|\mathbf{z} - \mathcal{A}\pi_h\|_2^2$$

$$h^\star = \min_{h \in \mathcal{H}_{k,\varepsilon,M}} R_{\pi_0}(h)$$

If *m large enough*

Trade-off with ε_0 minimal separation

w.h.p. $R_{\pi_0}(\hat{h}) - R_{\pi_0}(h^\star) \lesssim \sqrt{D_{KL}(\pi_0 || \mathcal{H}_{k,\varepsilon,M})} + \mathcal{O}\left(\sqrt{1/n}\right)$

GMM trade-off

Separation of means

Size of sketch

More high frequencies

Separation of means	Number of measurements
$\mathcal{O}\left(\sqrt{d \log k}\right)$	$m \geq \mathcal{O}\left(k^2 d^2 \cdot \texttt{polylog}(k,d)\right)$
$\mathcal{O}\left(\sqrt{d + \log k}\right)$	$m \geq \mathcal{O}\left(k^3 d^2 \cdot \texttt{polylog}(k,d)\right)$
$\mathcal{O}\left(\sqrt{\log k}\right)$	$m \geq \mathcal{O}\left(k^2 d^2 e^d \cdot \texttt{polylog}(k,d)\right)$

Sketch Size

Non-convex optimization.
Greedy heuristic: CL-OMP
[Keriven 2016]

In theory, at least

$$m \geq \mathcal{O}(k^2 d^2)$$

Empirically

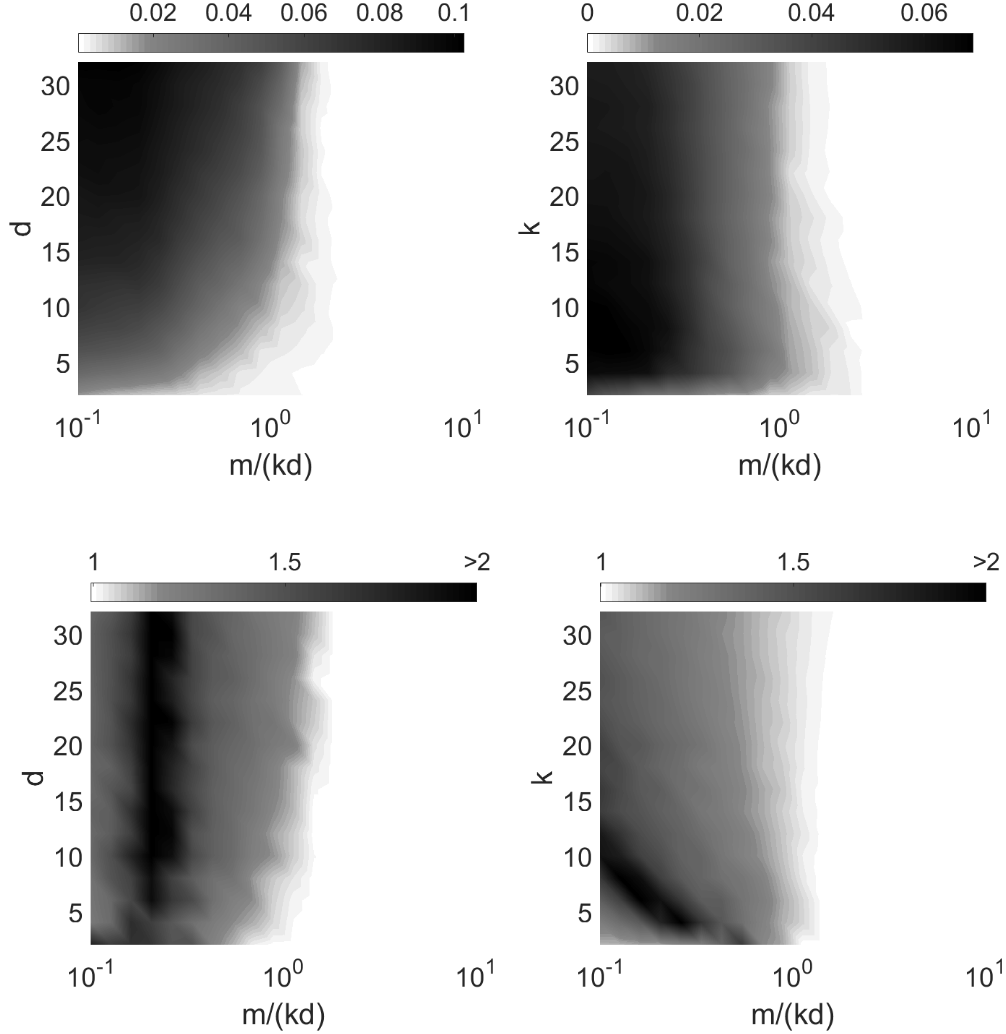
$$m \approx \mathcal{O}(kd)$$

k-means

SSE on k points

GMMs, known cov.

Relative loglike



Sketch of proof

Key idea 1

Sketching operator =

Kernel mean embedding *[Smola 2007]*

+ Random Features *[Rahimi 2007]*

Step 1

Relate risk to kernel metric

Sketch of proof

Key idea 1

Sketching operator =

Kernel mean embedding *[Smola 2007]*

+ Random Features *[Rahimi 2007]*

Key idea 2

Compressive Sensing analysis

[Bourrier 2014]

Step 1

Relate risk to kernel metric

Step 2

$\mathcal{A}$ satisfies the RIP

Sketch of proof

Key idea 1

Sketching operator =

Kernel mean embedding *[Smola 2007]*

\+ Random Features *[Rahimi 2007]*

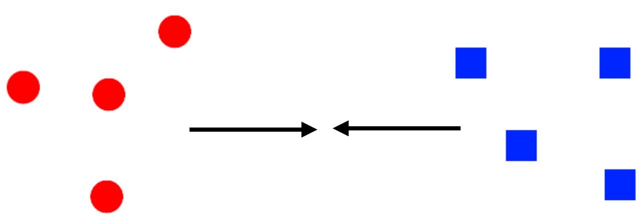
Key idea 2

Compressive Sensing analysis

[Bourrier 2014]

Main difficulty

*Controlling metrics between **mixtures** that get **close to each other** in infinite-dimensional space*



$$\|\sum_l \alpha_l \pi_l - \sum_l \alpha'_l \pi'_l\| \to 0 : \text{ what happens ?}$$

Step 1

Relate risk to kernel metric

Step 2

$\mathcal{A}$ satisfies the RIP

Sketch of proof

Key idea 1

Sketching operator =

Kernel mean embedding *[Smola 2007]*

+ Random Features *[Rahimi 2007]*

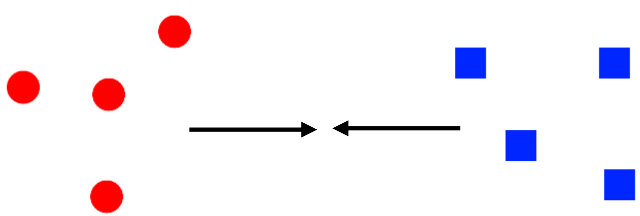
Key idea 2

Compressive Sensing analysis

[Bourrier 2014]

Main difficulty

*Controlling metrics between **mixtures** that get **close to each other** in infinite-dimensional space*



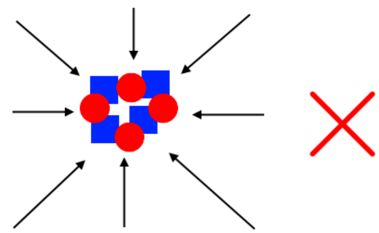
$$\|\sum_l \alpha_l \pi_l - \sum_l \alpha'_l \pi'_l\| \to 0 : \text{ what happens ?}$$

Step 1

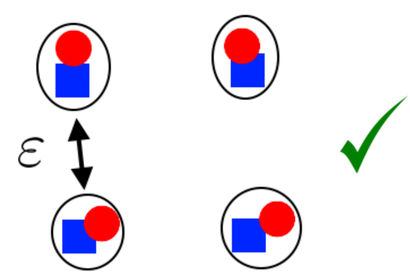
Relate risk to kernel metric

Step 2

$\mathcal{A}$ satisfies the RIP



No hypothesis



Separation hypothesis

Outline

1. Introduction
2. Main results
3. Experimental illustration
4. **Conclusion**

Conclusions

Contributions

- Efficient **sketched mixture learning** framework, using **random generalized moments**
- Combination of many tools:
  - Kernel mean embedding
  - Random Fourier features
  - Analysis inspired by Compressive Sensing

Conclusions

Contributions

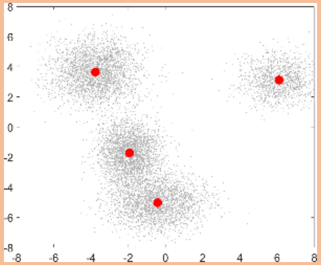
- Efficient **sketched mixture learning** framework, using **random generalized moments**
- Combination of many tools:
  - Kernel mean embedding
  - Random Fourier features
  - Analysis inspired by Compressive Sensing

Outlooks

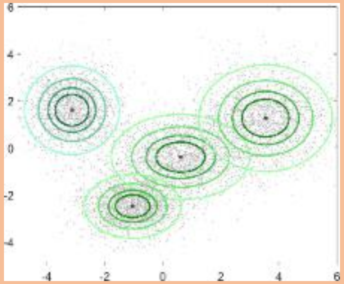
- Bridge gap theory / practice
- Other models (done in practice), with other sketching operators
- Non-linear sketches ? (neural networks…)

The SketchMLbox

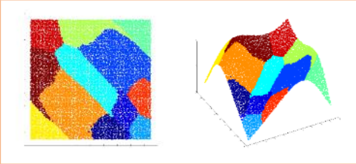
SketchMLbox (`sketchml.gforge.inria.fr`)



- Mixture of Diracs (« K-means »)
- GMMs with known covariance
- **GMMs with unknown diagonal covariance**



- **Soon**:
  - **Mixtures of multivariate alpha-stable (only known algorithm !)**
  - Gaussian Locally Linear Mapping *[Deleforge 2014]*



- **Optimized for user-defined**

Thank you !

- K., Bourrier, Gribonval, Perez. **Sketching for Large-Scale Learning of Mixture Models** *ICASSP 2016*
- K., Bourrier, Gribonval, Perez. **Sketching for Large-Scale Learning of Mixture Models** (extended version) *submitted to Information and Inference, arXiv:1606.0238*
- K., Tremblay, Gribonval, Traonmilin. **Compressive K-means** *ICASSP 2017*
- K., Tremblay, Gribonval. **SketchMLbox** (`sketchml.gforge.inria.fr`)
- Gribonval, Blanchard, K., Traonmilin. **Compressive Statistical Learning** ***online soon***

Appendix : CLOMPR

Algorithm 2: Compressive mixture learning *à la* OMP: CLOMP ($T = K$) and CLOMPR ($T = 2K$)

Data: Empirical sketch $\hat{\mathbf{z}}$, sketching operator $\mathcal{A}$, sparsity K, number of iterations $T \geq K$
Result: Support Θ, weights $\boldsymbol{\alpha}$
$\hat{\mathbf{r}} \leftarrow \hat{\mathbf{z}}$; $\Theta \leftarrow \emptyset$;
for $t \leftarrow 1$ **to** T **do**
  Step 1: Find a normalized atom highly correlated with the residual with a gradient descent
    $\boldsymbol{\theta} \leftarrow \texttt{maximize}_{\boldsymbol{\theta}} \left(\mathrm{Re} \left\langle \frac{\mathcal{A}P_{\boldsymbol{\theta}}}{\|\mathcal{A}P_{\boldsymbol{\theta}}\|_2}, \hat{\mathbf{r}} \right\rangle_2, \texttt{init} = \texttt{rand} \right)$;
  end
  Step 2: Expand support
    $\Theta \leftarrow \Theta \cup \{\boldsymbol{\theta}\}$;
  end
  Step 3: Enforce sparsity by Hard Thresholding if needed
    if $|\Theta| > K$ **then**
      $\boldsymbol{\beta} \leftarrow \arg\min_{\boldsymbol{\beta} \geq 0} \left\| \hat{\mathbf{z}} - \sum_{k=1}^{|\Theta|} \beta_k \frac{\mathcal{A}P_{\boldsymbol{\theta}_k}}{\|\mathcal{A}P_{\boldsymbol{\theta}_k}\|_2} \right\|_2$ Select K largest entries $\beta_{i_1}, ..., \beta_{i_K}$;
      Reduce the support $\Theta \leftarrow \{\boldsymbol{\theta}_{i_1}, ..., \boldsymbol{\theta}_{i_K}\}$;
    end
  end
  Step 4: Project to find weights
    $\boldsymbol{\alpha} \leftarrow \arg\min_{\boldsymbol{\alpha} \geq 0} \left\| \hat{\mathbf{z}} - \sum_{k=1}^{|\Theta|} \alpha_k \mathcal{A}P_{\boldsymbol{\theta}_k} \right\|_2$;
  end
  Step 5: Perform a gradient descent *initialized with current parameters*
    $\Theta, \boldsymbol{\alpha} \leftarrow \texttt{minimize}_{\Theta, \boldsymbol{\alpha}} \left(\left\| \hat{\mathbf{z}} - \sum_{k=1}^{|\Theta|} \alpha_k \mathcal{A}P_{\boldsymbol{\theta}_k} \right\|_2, \texttt{init} = (\Theta, \boldsymbol{\alpha}), \texttt{constraint} = \{\boldsymbol{\alpha} \geq 0\} \right)$;
  end
  Update residual: $\hat{\mathbf{r}} \leftarrow \hat{\mathbf{z}} - \sum_{k=1}^{|\Theta|} \alpha_k \mathcal{A}P_{\boldsymbol{\theta}_k}$
end
Normalize $\boldsymbol{\alpha}$ such that $\sum_{k=1}^{K} \alpha_k = 1$