Adaptive two-sample testing

Arthur Gretton

Gatsby Computational Neuroscience Unit,
Google Deepmind

Cambridge, 2023

Comparing two samples

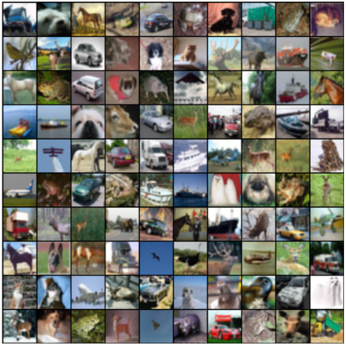
- Given: Samples from unknown distributions P and Q.
- Goal: do P and Q differ?

$$\sim P$$

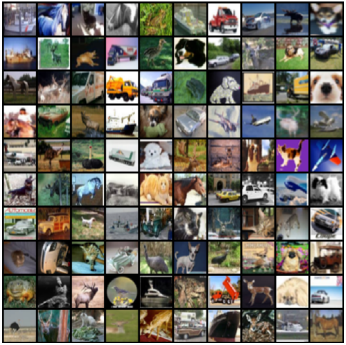
$$\sim Q$$

A real-life example: two-sample tests

- Goal: do P and Q differ?



CIFAR 10 samples



Cifar 10.1 samples

Significant difference?

Feng, Xu, Lu, Zhang, G., Sutherland, Learning Deep Kernels for Non-Parametric Two-Sample Tests, ICML 2020

Sutherland, Tung, Strathmann, De, Ramdas, Smola, G., ICLR 2017.

Two-sample problem

- Samples $\mathbb{X}_m := (x_1, \ldots, x_m)$, $x_i \overset{\text{iid}}{\sim} p$ in $\mathbb{R}^d$
- Samples $\mathbb{Y}_n := (y_1, \ldots, y_n)$, $y_i \overset{\text{iid}}{\sim} q$ in $\mathbb{R}^d$

where $m \leq n$ and $n \leq Cm$.

Hypothesis test: function $\Delta_\alpha(\mathbb{X}_m, \mathbb{Y}_n)$

$\mathcal{H}_0 \colon p = q$	against	$\mathcal{H}_1 \colon p \neq q$
$\Delta_\alpha(\mathbb{X}_m, \mathbb{Y}_n) = 1$	$\iff$	reject $\mathcal{H}_0$
$\Delta_\alpha(\mathbb{X}_m, \mathbb{Y}_n) = 0$	$\iff$	fail to reject $\mathcal{H}_0$

Type II error β

$$\mathbb{P}_{p \times q}(\Delta(\mathbb{X}_m, \mathbb{Y}_n) = 0) \;\leq\; \beta$$

Type I error: controlled by α by design

$$\mathbb{P}_{p \times p}(\Delta(\mathbb{X}_m, \mathbb{Y}_n) = 1) \;\leq\; \alpha$$

Outline

Two sample testing

- Test statistic: Maximum Mean Discrepancy (MMD)...
  - ...as a difference in feature means
  - ...as an integral probability metric
- Statistical testing with the MMD
- "How to choose the best kernel"
  - using aggregation (no sample splitting)
  - minimax guarantees with Sobolev smoothness assumption

Maximum Mean Discrepancy

Kernel methods, feature representation

Kernels: dot products of features

Feature map $\varphi(x) \in \mathcal{F}$,

$$\varphi(x) = [\ldots \varphi_i(x) \ldots] \in \ell_2$$

For positive definite k,

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{F}}$$

Infinitely many features $\varphi(x)$, dot product in closed form!

Kernel methods, feature representation

Kernels: dot products of features

Feature map $\varphi(x) \in \mathcal{F}$,

$$\varphi(x) = [\ldots \varphi_i(x) \ldots] \in \ell_2$$

For positive definite k,

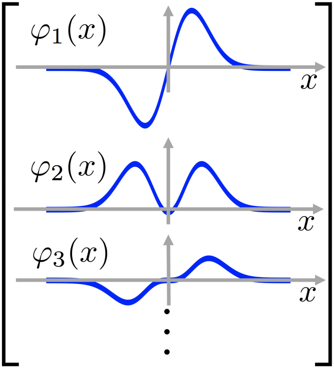
$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{F}}$$

Infinitely many features $\varphi(x)$, dot product in closed form!

Exponentiated quadratic kernel

$$k(x, x') = \exp\left(-\gamma \|x - x'\|^2\right)$$

$\varphi(x) =$



Features: Gaussian Processes for Machine learning, Rasmussen and Williams, Ch. 4.

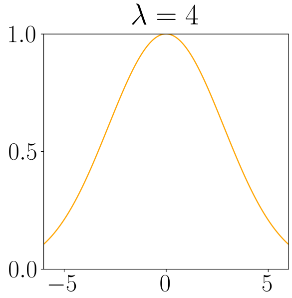
Kernels and bandwidths

Kernel: $k_\lambda(x, y) := \prod_{i=1}^{d} \frac{1}{\lambda_i} K_i\left(\frac{x_i - y_i}{\lambda_i}\right)$ Bandwidth: $\lambda \in (0, \infty)^d$

Assumptions: $K_1, \ldots, K_d$ integrable (to 1) and square integrable

Eg: Gaussian ($K_i(u) \propto e^{-u^2}$), Laplace ($K_i(u) \propto e^{-|u|}$), Matérn,...

Gaussian kernel: $k_\lambda(x - y) := \frac{1}{(\pi)^{d/2}} \prod_{i=1}^{d} \frac{1}{\lambda_i} \exp\left(-\frac{(x_i - y_i)^2}{\lambda_i{}^2}\right)$



Kernel mean embedding

Function evaluation in an RKHS:

$$f(x) = \langle f, \varphi_x \rangle_{\mathcal{F}}$$

Expectation evaluation in an RKHS:

$$\mathrm{E}_P(f(X)) = \mathrm{E}_P \langle f, \varphi_X \rangle_{\mathcal{F}} = \langle f, \mathrm{E}_P \varphi_X \rangle_{\mathcal{F}} =: \langle f, \mu_P \rangle_{\mathcal{F}}$$

as long as feature map Bochner integrable: $\mathrm{E}_P \|\varphi_X\| = \mathrm{E}_P \sqrt{k_\lambda(X, X)} < \infty$.

μ_P gives you expectations of all RKHS functions

"Kernel trick" for mean embeddings:

$$\langle \mu_P, \mu_Q \rangle_{\mathcal{F}} = \mathrm{E}_{P,Q} k_\lambda(X, Y)$$

for $X \sim P$ and $Y \sim Q$.

The maximum mean discrepancy

The maximum mean discrepancy is the distance between feature means:

$$\begin{aligned}\mathrm{MMD}^2_\lambda(p, q) &= \|\mu_P - \mu_Q\|^2_{\mathcal{F}} \\ &= \underbrace{\mathrm{E}_P k_\lambda(X, X')}_{\text{(a)}} + \underbrace{\mathrm{E}_Q k_\lambda(Y, Y')}_{\text{(a)}} - 2\underbrace{\mathrm{E}_{P,Q} k_\lambda(X, Y)}_{\text{(b)}}\end{aligned}$$

(a)= within distrib. similarity, (b)= cross-distrib. similarity.

Characteristic kernels on $\mathbb{R}^d$

Characteristic kernel: $\mathrm{MMD}^2_\lambda(p, q) = 0$ iff $p = q$ Fukumizu et al. [NIPS07b], Sriperumbudur et al.[COLT08]

When are translation invariant kernels $k_\lambda(x, y) = k_\lambda(x - y)$ characteristic on $\mathbb{R}^d$?

Characteristic kernels on $\mathbb{R}^d$

Characteristic kernel: $\mathrm{MMD}^2_\lambda(p, q) = 0$ iff $p = q$ Fukumizu et al. [NIPS07b], Sriperumbudur et al.[COLT08]

When are translation invariant kernels $k_\lambda(x, y) = k_\lambda(x - y)$ characteristic on $\mathbb{R}^d$?

Bochner's theorem:

$$k_\lambda(x - y) = \int_{\mathbb{R}^d} e^{-i(x-y)^\top \omega} d\Lambda(\omega)$$

$\Lambda(\omega)$ finite non-negative Borel measure.

Characteristic function of P via Fourier transform

$$\varphi_P(\omega) = \int_{\mathbb{R}^d} e^{ix^\top \omega} dP(x)$$

Characteristic kernels on $\mathbb{R}^d$

Fourier representation of MMD on $\mathbb{R}^d$:

$$\mathrm{MMD}_\lambda^2(p, q) = \int \left|\varphi_P(\omega) - \varphi_Q(\omega)\right|^2 d\Lambda(\omega)$$

Proof:

$$\begin{aligned}
&\mathrm{MMD}_\lambda^2(p, q) \\
&:= E_P k_\lambda(x - x') + E_Q k_\lambda(y - y') - 2E_{P,Q} k_\lambda(x, y) \\
&= \int\int \left[k_\lambda(s - t)\, d(P - Q)(s)\right] d(P - Q)(t) \\
&= \int_{\mathbb{R}^d} \left|\phi_P(\omega) - \phi_Q(\omega)\right|^2 \, d\Lambda(\omega)
\end{aligned}$$

Characteristic kernels on $\mathbb{R}^d$

Fourier representation of MMD on $\mathbb{R}^d$:

$$\mathrm{MMD}^2_\lambda(p, q) = \int |\varphi_P(\omega) - \varphi_Q(\omega)|^2 \, d\Lambda(\omega)$$

Proof:

$$\begin{aligned}
&\mathrm{MMD}^2_\lambda(p, q) \\
&:= E_P k_\lambda(x - x') + E_Q k_\lambda(y - y') - 2E_{P,Q} k_\lambda(x, y) \\
&= \int \int \left[k_\lambda(s - t) \, d(P - Q)(s) \right] d(P - Q)(t) \\
&= \int_{\mathbb{R}^d} |\phi_P(\omega) - \phi_Q(\omega)|^2 \, d\Lambda(\omega)
\end{aligned}$$

Characteristic kernels on $\mathbb{R}^d$

Fourier representation of MMD on $\mathbb{R}^d$:

$$\mathrm{MMD}^2_\lambda(p, q) = \int \left|\varphi_P(\omega) - \varphi_Q(\omega)\right|^2 d\Lambda(\omega)$$

Proof:

$$\begin{aligned}
&\mathrm{MMD}^2_\lambda(p, q) \\
&:= E_P k_\lambda(x - x') + E_Q k_\lambda(y - y') - 2E_{P,Q} k_\lambda(x, y) \\
&= \int\int \left[k_\lambda(s - t)\, d(P - Q)(s)\right] d(P - Q)(t) \\
&= \int_{\mathbb{R}^d} \left|\phi_P(\omega) - \phi_Q(\omega)\right|^2 \, d\Lambda(\omega)
\end{aligned}$$

Summary: characteristic kernels on $\mathbb{R}^d$

A translation invariant k_λ is characteristic for prob. measures on $\mathbb{R}^d$ if and only if

$$\mathrm{supp}(\Lambda) = \mathbb{R}^d$$

(i.e. support zero on at most a countable set) Sriperumbudur et al. [COLT08, JMLR10]

Corollary: any continuous, compactly supported k_λ characteristic (since Fourier spectrum $\Lambda(\omega)$ cannot be zero on an interval).

1-D proof sketch from [Mallat, 99, Theorem 2.6], proof on $\mathbb{R}^d$ via distribution theory in Sriperumbudur et al. [JMLR10, Corollary 10 p. 1535]

Summary: characteristic kernels on $\mathbb{R}^d$

A translation invariant k_λ is characteristic for prob. measures on $\mathbb{R}^d$ if and only if

$$\operatorname{supp}(\Lambda) = \mathbb{R}^d$$

(i.e. support zero on at most a countable set) Sriperumbudur et al. [COLT08, JMLR10]

Corollary: any continuous, compactly supported k_λ characteristic (since Fourier spectrum $\Lambda(\omega)$ cannot be zero on an interval).

1-D proof sketch from [Mallat, 99, Theorem 2.6], proof on $\mathbb{R}^d$ via distribution theory in Sriperumbudur et al. [JMLR10, Corollary 10 p. 1535]

Two-Sample Testing with MMD

A statistical test using MMD

The empirical MMD:

$$\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n) = \frac{1}{m(m-1)} \sum_{i \neq j} k_\lambda(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k_\lambda(\mathrm{y}_i, \mathrm{y}_j) - \frac{2}{mn} \sum_{i,j} k_\lambda(x_i, \mathrm{y}_j)$$

Two-sample MMD test:

$$\Delta^\lambda_\alpha(\mathbb{X}_m, \mathbb{Y}_n) := \mathbb{1}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n) > q^\lambda_{1-\alpha}\right)$$

A statistical test using MMD

The empirical MMD:

$$\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n) = \frac{1}{m(m-1)} \sum_{i \neq j} k_\lambda(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k_\lambda(\mathrm{y}_i, \mathrm{y}_j) - \frac{2}{mn} \sum_{i,j} k_\lambda(x_i, \mathrm{y}_j)$$

Two-sample MMD test:

$$\Delta^\lambda_\alpha(\mathbb{X}_m, \mathbb{Y}_n) := \mathbb{1}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n) > q^\lambda_{1-\alpha}\right)$$

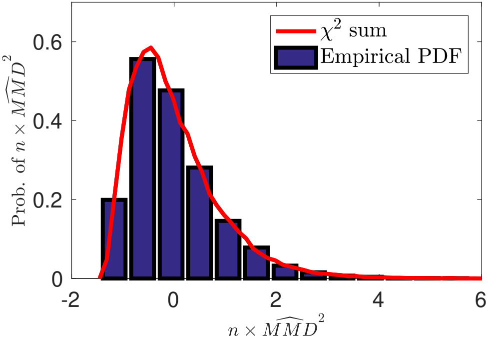
Want threshold $q^\lambda_{1-\alpha}$ for test $\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n)$ to get false positive rate α

Asymptotics of $\widehat{MMD}^2$ when $P = Q$

$P = Q = \mathcal{N}(0,1)$, statistic has asymptotic distribution

$$(m+n)\widehat{\text{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n) \sim \sum_{l=1}^{\infty} \lambda_l \left[z_l^2 - 2\right]$$

MMD density under $\mathcal{H}_0$



where

$$\lambda_i \psi_i(x') = \int_{\mathcal{X}} \underbrace{\tilde{k}_\lambda(x, x')}_{\text{centred}} \psi_i(x)\, dP(x)$$

$$z_l \sim \mathcal{N}(0, 2) \quad \text{i.i.d.}$$

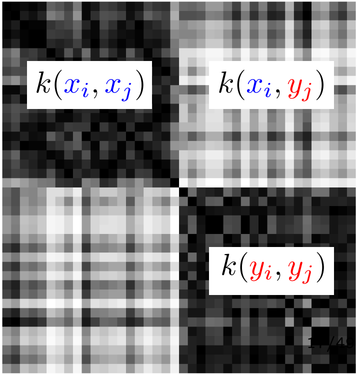
How do we get the test threshold $q_{1-\alpha}^{\lambda}$?

Original empirical MMD for dogs and fish:

$X = [$  $\dots]$

$Y = [$  $\dots]$

$$\widehat{MMD}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(y_i, y_j) - \frac{2}{n^2} \sum_{i,j} k(x_i, y_j)$$

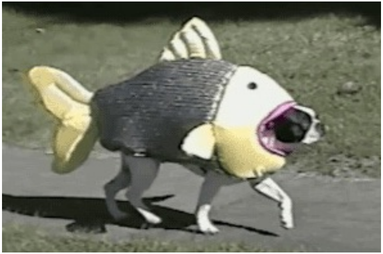


How do we get test threshold $q^{\lambda}_{1-\alpha}$?

Permuted dog and fish samples (merdogs):

$\widetilde{X} = [$  $\ldots\ ]$

$\widetilde{Y} = [$  $\ldots\ ]$



How do we get test threshold $q^{\lambda}_{1-\alpha}$?

Permuted dog and fish samples (merdogs):

$\widetilde{X} = [$  $\dots]$

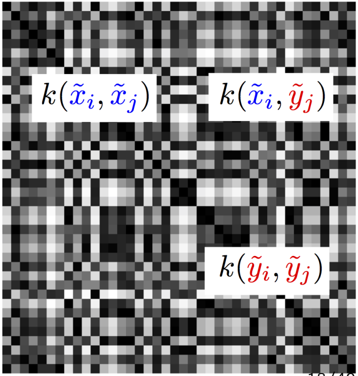
$\widetilde{Y} = [$  $\dots]$

$$\widehat{\mathrm{MMD}}^2_{\lambda}(\mathbb{X}_m, \mathbb{Y}_n)$$

$$\frac{1}{m(m-1)} \sum_{i \neq j} k(\tilde{\mathrm{x}}_i, \tilde{\mathrm{x}}_j)$$

$$+ \frac{1}{n(n-1)} \sum_{i \neq j} k(\tilde{\mathrm{y}}_i, \tilde{\mathrm{y}}_j)$$

$$- \frac{2}{mn} \sum_{i,j} k(\tilde{\mathrm{x}}_i, \tilde{\mathrm{y}}_j)$$



MMD test thresholds: permutation, wild bootstrap

Two-sample MMD test:

$$\Delta_{\alpha}^{\lambda, B}(\mathbb{X}_m, \mathbb{Y}_n) := \mathbb{1}\left(\widehat{\mathrm{MMD}}_{\lambda}^{2}(\mathbb{X}_m, \mathbb{Y}_n) > \widehat{q}_{1-\alpha}^{\lambda}\right)$$

Quantile: $\widehat{q}_{1-\alpha}^{\lambda}$ is the $\lceil (B+1)(1-\alpha) \rceil$-th largest value of $\widehat{\mathrm{MMD}}_{\lambda}^{2}(\mathbb{X}_m, \mathbb{Y}_n)$ and B $\mathcal{H}_0$-simulated test statistics

Permutations: $\widehat{\mathrm{MMD}}_{\lambda}^{2}(\mathbb{X}_m^{\sigma}, \mathbb{Y}_n^{\sigma})$ where $(\mathbb{X}_m^{\sigma}, \mathbb{Y}_n^{\sigma}) = \sigma(\mathbb{X}_m \cup \mathbb{Y}_n)$

MMD test thresholds: permutation, wild bootstrap

Two-sample MMD test:

$$\Delta_\alpha^{\lambda,B}(\mathbb{X}_m, \mathbb{Y}_n) := \mathbb{1}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n) > \widehat{q}^{\,\lambda}_{1-\alpha}\right)$$

Quantile: $\widehat{q}^{\,\lambda}_{1-\alpha}$ is the $\lceil (B+1)(1-\alpha) \rceil$-th largest value of $\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n)$ and B $\mathcal{H}_0$-simulated test statistics

Permutations: $\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}^\sigma_m, \mathbb{Y}^\sigma_n)$ where $(\mathbb{X}^\sigma_m, \mathbb{Y}^\sigma_n) = \sigma(\mathbb{X}_m \cup \mathbb{Y}_n)$

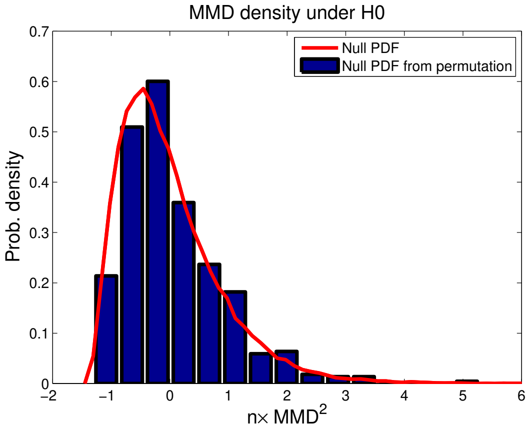
Non-asymptotic level (permutation): $\mathbb{P}_{p\times p}\left(\Delta_\alpha^{\lambda,B}(\mathbb{X}_m, \mathbb{Y}_n) = 1\right) \ \leq \ \alpha,$

Time complexity: $\mathcal{O}\left(B\,(m+n)^2\right)$

Approx. null distribution of $\widehat{MMD}^2$ via permutation

Null distribution estimated from 500 permutations

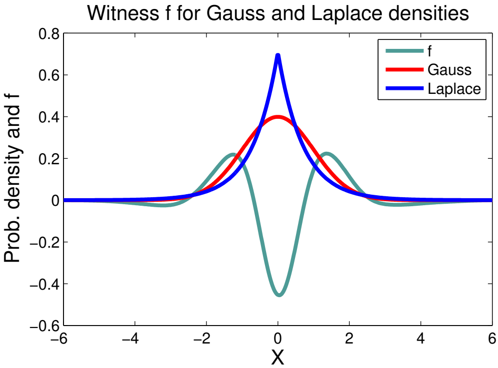
Example: $P = Q = \mathcal{N}(0,1)$



Kernel choice: MMD as an IPM

Maximum mean discrepancy: smooth function for P vs Q

$$\mathrm{MMD}_{\lambda}(p, q) := \sup_{\|f\|_{\mathcal{F}} \leq 1} \left[\mathrm{E}_P f(X) - \mathrm{E}_Q f(Y) \right]$$



Kernel Choice: MMD as an IPM

- Simple choice: Gaussian

$$k_{\lambda}(x, y) = \frac{1}{(\pi)^{d/2}} \prod_{i=1}^{d} \frac{1}{\lambda_i} \exp\left(-\frac{(x_i - y_i)^2}{\lambda_i{}^2} \right)$$

- *Characteristic:* for any σ: for any P and Q, power $\to 1$ as $n \to \infty$

Kernel Choice: MMD as an IPM

- Simple choice: Gaussian

$$k_{\lambda}(x,y) = \frac{1}{(\pi)^{d/2}} \prod_{i=1}^{d} \frac{1}{\lambda_i} \exp\left(- \frac{(x_i - y_i)^2}{{\lambda_i}^2} \right)$$

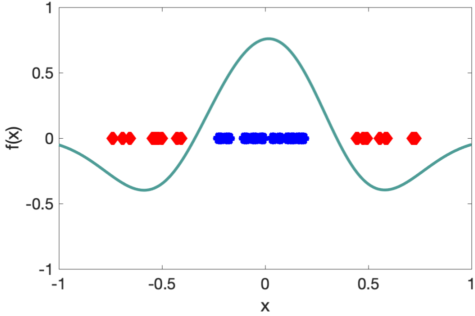
- *Characteristic:* for any σ: for any P and Q, power $\to 1$ as $n \to \infty$
- But choice of $\lambda_1 \cdots \lambda_d$ is very important for finite $m, n \ldots$

Kernel Choice: MMD as an IPM

- Simple choice: Gaussian

$$k_{\lambda}(x, y) = \frac{1}{(\pi)^{d/2}} \prod_{i=1}^{d} \frac{1}{\lambda_i} \exp\left(-\frac{(x_i - y_i)^2}{\lambda_i{}^2} \right)$$

- *Characteristic:* for any σ: for any P and Q, power $\to 1$ as $n \to \infty$
- But choice of $\lambda_1 \cdots \lambda_d$ is very important for finite $m, n \ldots$

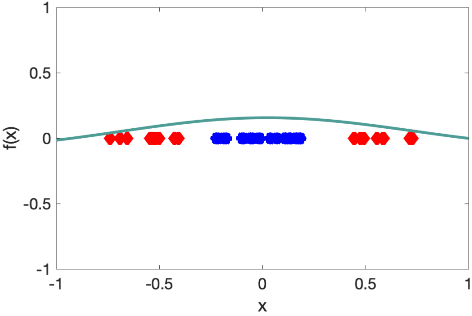


Kernel Choice: MMD as an IPM

- Simple choice: Gaussian

$$k_{\lambda}(x, y) = \frac{1}{(\pi)^{d/2}} \prod_{i=1}^{d} \frac{1}{\lambda_i} \exp\left(-\frac{(x_i - y_i)^2}{\lambda_i{}^2}\right)$$

- *Characteristic:* for any σ: for any P and Q, power $\to 1$ as $n \to \infty$
- But choice of $\lambda_1 \cdots \lambda_d$ is very important for finite $m, n \ldots$

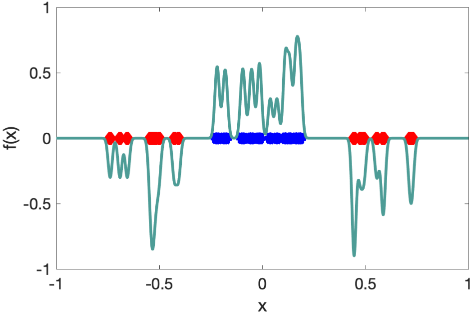


Kernel Choice: MMD as an IPM

- Simple choice: Gaussian

$$k_\lambda(x,y) = \frac{1}{(\pi)^{d/2}} \prod_{i=1}^{d} \frac{1}{\lambda_i} \exp\left(-\frac{(x_i - y_i)^2}{{\lambda_i}^2} \right)$$

- *Characteristic:* for any σ: for any P and Q, power $\to 1$ as $n \to \infty$
- But choice of $\lambda_1 \cdots \lambda_d$ is very important for finite $m, n\ldots$



Test power for known smoothness of $p - q$

Sobolev balls

Regularity/smoothness assumption: $p - q \in \mathcal{S}_d^s(R)$

Sobolev balls:

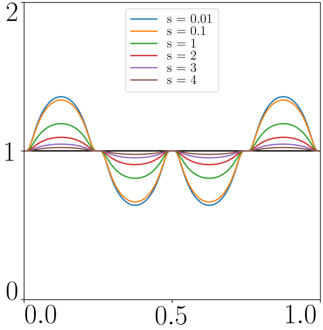
$$\mathcal{S}_d^s(R) := \left\{ f \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d) : \int_{\mathbb{R}^d} \|\xi\|_2^{2s} |\widehat{f}(\xi)|^2 \, \mathrm{d}\xi \leq (2\pi)^d R^2 \right\}$$

radius $R > 0$

dimension d

smoothness parameter $s > 0$

Fourier transform $\widehat{f}(\xi) := \int_{\mathbb{R}^d} f(x) e^{-ix^\top \xi} \mathrm{d}x$



Sobolev balls

Regularity/smoothness assumption: $p - q \in \mathcal{S}_d^s(R)$

Sobolev balls:

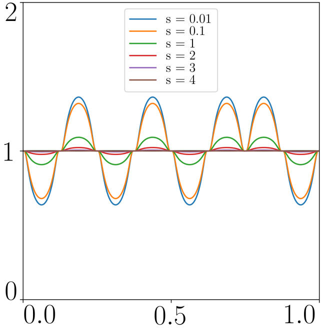
$$\mathcal{S}_d^s(R) := \left\{ f \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d) : \int_{\mathbb{R}^d} \|\xi\|_2^{2s} |\widehat{f}(\xi)|^2 \, \mathrm{d}\xi \le (2\pi)^d R^2 \right\}$$

radius $R > 0$

dimension d

smoothness parameter $s > 0$

Fourier transform $\widehat{f}(\xi) := \int_{\mathbb{R}^d} f(x) e^{-ix^\top \xi} \mathrm{d}x$



MMD test power, *known* smoothness

Theorem (MMD test minimax optimality)

For known smoothness s, assuming $p - q \in \mathcal{S}_d^s(R)$ and setting

$$\lambda_i^\star := (m+n)^{-2/(4s+d)}$$

for $i = 1, \ldots, d$, the condition

$$\|p-q\|_2 \;\geq\; \frac{C}{\sqrt{\beta}}(m+n)^{-2s/(4s+d)}$$

guarantees control of the type II error of the MMD test

$$\mathbb{P}_{p\times q\times r}\left(\Delta_\alpha^{\lambda,B}(\mathbb{X}_m, \mathbb{Y}_n) = 0\right) \;\leq\; \beta.$$

Minimax rate over Sobolev balls: $(m+n)^{-2s/(4s+d)}$

Proof of theorem 1 (next few slides)

For $\alpha, \beta \in (0,1)$, Type II error control

$$\mathbb{P}_{p \times q \times r}\left(\Delta_{\alpha}^{\lambda, B}(\mathbb{X}_m, \mathbb{Y}_n) = 0\right) \leq \beta$$

is implied by (Chebyshev)

$$\mathbb{P}_{p \times q \times r}\left(\mathrm{MMD}_{\lambda}^2(p, q) \geq \sqrt{\frac{2}{\beta}\mathrm{var}_{p \times q}\left(\widehat{\mathrm{MMD}}_{\lambda}^2(\mathbb{X}_m, \mathbb{Y}_n)\right)} + \widehat{q}_{1-\alpha}^{\lambda}\right) \geq 1 - \frac{\beta}{2}$$

Proof of theorem 1 (next few slides)

$$\underbrace{\mathrm{MMD}^2_\lambda(p, q)}_{(A)} \geq \underbrace{\sqrt{\frac{2}{\beta}\mathrm{var}_{p\times q}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n)\right)}}_{(B)} + \underbrace{\widehat{q}^{\,\lambda}_{1-\alpha}}_{(C)}$$

We address each of the three terms (A), (B), (C) in turn.

Breakdown of the MMD (A)

The MMD can be decomposed

$$\begin{aligned}\mathrm{MMD}_\lambda^2(p, q) &= \langle p - q, k_\lambda * (p - q)\rangle_2 \\ &= \frac{1}{2}\Big(\|p - q\|_2^2 + \|k_\lambda * (p - q)\|_2^2 \\ &\qquad - \|k_\lambda * (p - q) - (p - q)\|_2^2\Big)\end{aligned}$$

Breakdown of the MMD (A)

The MMD can be decomposed

$$\begin{aligned}\mathrm{MMD}^2_\lambda(p, q) &= \langle p - q, k_\lambda * (p - q)\rangle_2 \\ &= \frac{1}{2}\Big(\|p - q\|_2^2 + \|k_\lambda * (p - q)\|_2^2 \\ &\qquad - \|k_\lambda * (p - q) - (p - q)\|_2^2\Big)\end{aligned}$$

- Keep the first term (test "radius" for power $1 - \beta$): $\|p - q\|_2^2$
- Get rid of second term using variance (next slides): $\|k_\lambda * (p - q)\|_2^2$
- Bound the final term:if $p - q \in \mathcal{S}_d^s(R)$, then $\exists S \in (0, 1)$ such that

$$\|k_\lambda * (p - q) - (p - q)\|_2^2 - S^2\|p - q\|_2^2 \leq C_0(d, s, R)\sum_{i=1}^{d} {\lambda_i}^{2s}$$

Updating the power condition after (A)

The power condition (which needs to hold with probability $1 - \beta/2$)

$$\underbrace{\mathrm{MMD}^2_{\lambda}(p, q)}_{(A)} \geq \underbrace{\sqrt{\frac{2}{\beta}\mathrm{var}_{p \times q}\left(\widehat{\mathrm{MMD}}^2_{\lambda}(\mathbb{X}_m, \mathbb{Y}_n)\right)}}_{(B)} + \underbrace{\widehat{q}^{\lambda}_{1-\alpha}}_{(C)},$$

Updating the power condition after (A)

The power condition (which needs to hold with probability $1 - \beta/2$)

$$\underbrace{\mathrm{MMD}^2_\lambda(p, q)}_{(A)} \;\geq\; \underbrace{\sqrt{\frac{2}{\beta}\mathrm{var}_{p\times q}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n)\right)}}_{(B)} + \underbrace{\widehat{q}^{\,\lambda}_{1-\alpha}}_{(C)},$$

after updating (A), then becomes

$$(1 - S^2)\|p - q\|_2^2 \;\geq\; C_0 \sum_{i=1}^{d} {\lambda_i}^{2s} - \|k_\lambda * (p - q)\|_2^2$$
$$+ 2\underbrace{\sqrt{\frac{2}{\beta}\mathrm{var}_{p\times q}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n)\right)}}_{(B)} + 2\underbrace{\widehat{q}^{\,\lambda}_{1-\alpha}}_{(C)}$$

Bound on the variance (B)

Assume that $\max\left(\|p\|_\infty, \|q\|_\infty\right) \le M$ for some $M > 0$.

$$\begin{aligned}
&\mathrm{var}_{p\times q}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n)\right)\\
&\le C_1(M,d)\left(\frac{\|k_\lambda * (p-q)\|_2^2}{m+n} + \frac{1}{(m+n)^2\lambda_1\cdots\lambda_d}\right).
\end{aligned}$$

Bound on the variance (B)

Assume that $\max(\|p\|_\infty, \|q\|_\infty) \le M$ for some $M > 0$.

$$\begin{aligned}
&\mathrm{var}_{p\times q}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n)\right) \\
&\le C_1(M,d)\left(\frac{\|k_\lambda * (p-q)\|_2^2}{m+n} + \frac{1}{(m+n)^2\lambda_1\cdots\lambda_d}\right).
\end{aligned}$$

Assuming $\lambda_1\cdots\lambda_d \le 1$,

$$\begin{aligned}
(B) =& 2\sqrt{\frac{2}{\beta}\mathrm{var}_{p\times q}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n)\right)} \\
&\le \|k_\lambda * (p-q)\|_2^2 + \frac{C_1'}{\sqrt{\beta}(m+n)\sqrt{\lambda_1\cdots\lambda_d}}.
\end{aligned}$$

Term $\|k_\lambda * (p-q)\|_2^2$ will cancel in the power condition.

Updating the power condition after (A), (B)

The power condition (which needs to hold with probability $1 - \beta/2$)

$$(1 - S^2)\|p - q\|_2^2 \;\geq\; C_0 \sum_{i=1}^{d} {\lambda_i}^{2s} - \|k_\lambda * (p - q)\|_2^2 + 2\underbrace{\sqrt{\frac{2}{\beta}\mathrm{var}_{p\times q}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n)\right)}}_{(B)} + 2\underbrace{\widehat{q}_{1-\alpha}^{\lambda}}_{(C)}.$$

Updating the power condition after (A), (B)

The power condition (which needs to hold with probability $1 - \beta/2$)

$$(1 - S^2)\|p - q\|_2^2 \geq C_0 \sum_{i=1}^{d} \lambda_i^{2s} - \|k_\lambda * (p - q)\|_2^2$$

$$+ 2\underbrace{\sqrt{\frac{2}{\beta}\mathrm{var}_{p \times q}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n)\right)}}_{(B)} + 2\underbrace{\widehat{q}_{1-\alpha}^{\lambda}}_{(C)}.$$

After bounding (B), becomes

$$(1 - S^2)\|p - q\|_2^2 \geq \frac{C_1'}{\sqrt{\beta}(m + n)\lambda_1 \cdots \lambda_d} + C_0 \sum_{i=1}^{d} \lambda_i^{2s} + 2\underbrace{\widehat{q}_{1-\alpha}^{\lambda}}_{(C)}.$$

Bound on estimated $1-\alpha$ quantile (C)

[Schrab et al., 2023, Proposition 4]

Assume that $\max\left(\|p\|_\infty, \|q\|_\infty\right) \le M$ for some $M > 0$. We have

$$\mathbb{P}_{p\times q\times r}\left(\widehat{q}_{1-\alpha}^{\lambda} \le C_2(M,d)\frac{\ln\left(\frac{1}{\alpha}\right)}{\sqrt{\beta}(m+n)\sqrt{\lambda_1\cdots\lambda_d}}\right) \ge 1-\frac{\beta}{2}$$

for $B \ge \frac{3}{\alpha^2}\left(\ln\left(\frac{8}{\beta}\right)+\alpha(1-\alpha)\right)$ and $\alpha \in (0, 0.5)$.

Updating the power condition after (A), (B), (C)

The power condition (which needs to hold with probability $1-\beta/2$)

$$(1-S^2)\|p-q\|_2^2 \ \geq \frac{C_1'}{\sqrt{\beta}(m+n)\sqrt{\lambda_1\cdots\lambda_d}} + C_0\sum_{i=1}^{d}\lambda_i{}^{2s} + 2\underbrace{\widehat{q}^{\lambda}_{1-\alpha}}_{(C)}.$$

Fine print: $\alpha \in (0, e^{-1})$, $B \geq \frac{3}{\alpha^2}\left(\ln\left(\frac{8}{\beta}\right) + \alpha(1-\alpha)\right)$, and $\lambda_1\cdots\lambda_d \leq 1$.

Updating the power condition after (A), (B), (C)

The power condition (which needs to hold with probability $1-\beta/2$)

$$(1-S^2)\|p-q\|_2^2 \;\geq\; \frac{C_1'}{\sqrt{\beta}(m+n)\sqrt{\lambda_1\cdots\lambda_d}} + C_0\sum_{i=1}^d {\lambda_i}^{2s} + \underbrace{2\widehat{q}_{1-\alpha}^{\lambda}}_{(C)}.$$

After updating (C)

$$\|p-q\|_2^2 \;\geq\; \frac{C_4(M,d,s,S,R)}{\sqrt{\beta}}\left(\sum_{i=1}^d {\lambda_i}^{2s} + \frac{\ln\left(\frac{1}{\alpha}\right)}{(m+n)\sqrt{\lambda_1\cdots\lambda_d}}\right).$$

Fine print: $\alpha\in(0,e^{-1})$, $B\geq\frac{3}{\alpha^2}\left(\ln\left(\frac{8}{\beta}\right)+\alpha(1-\alpha)\right)$, and $\lambda_1\cdots\lambda_d\leq 1$.

Updating the power condition after (A), (B), (C)

The power condition (which needs to hold with probability $1 - \beta/2$)

$$(1 - S^2)\|p - q\|_2^2 \ \geq \ \frac{C_1'}{\sqrt{\beta}(m+n)\sqrt{\lambda_1 \cdots \lambda_d}} + C_0 \sum_{i=1}^{d} \lambda_i^{2s} + 2\underbrace{\widehat{q}^{\,\lambda}_{1-\alpha}}_{(C)}.$$

After updating (C)

$$\|p - q\|_2^2 \ \geq \ \frac{C_4(M, d, s, S, R)}{\sqrt{\beta}} \left(\sum_{i=1}^{d} \lambda_i^{2s} + \frac{\ln\left(\frac{1}{\alpha}\right)}{(m+n)\sqrt{\lambda_1 \cdots \lambda_d}} \right).$$

Picking $\lambda_i^\star := (m+n)^{-2/(4s+d)}$ controls the Type II error when

$$\|p - q\|_2 \ \geq \ \frac{C}{\sqrt{\beta}} (m+n)^{-2s/(4s+d)}$$

Fine print: $\alpha \in (0, e^{-1})$, $B \geq \frac{3}{\alpha^2}\left(\ln\left(\frac{8}{\beta}\right) + \alpha(1-\alpha)\right)$, and $\lambda_1 \cdots \lambda_d \leq 1$.

Optimizing kernel parameters: aggregation

MMDAgg for a *collection* of bandwidths Λ

MMDAgg (MMD Aggregation): non-asymptotic level α

$$\Delta_\alpha^\Lambda(\mathbb{X}_m, \mathbb{Y}_n) := \mathbb{1}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n) > \widehat{q}^{\,\lambda}_{1-u_\alpha w_\lambda} \text{ for some } \lambda \in \Lambda\right)$$

positive weights $(w_\lambda)_{\lambda\in\Lambda}$ satisfying $\sum_{\lambda\in\Lambda} w_\lambda \leq 1$

Correction u_α defined as

$$\sup\left\{u > 0 : \mathbb{P}_{p\times p}\left(\max_{\lambda\in\Lambda}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n) - \widehat{q}^{\,\lambda}_{1-uw_\lambda}\right) > 0\right) \leq \alpha\right\}$$

more powerful than Bonferroni correction as $u_\alpha \geq \alpha$

Time complexity $\mathcal{O}\left(|\Lambda|\,(B_1 + B_2)\,(m+n)^2\right)$

MMDAgg for a *collection* of bandwidths Λ

MMDAgg (MMD Aggregation): non-asymptotic level α

$$\Delta_\alpha^\Lambda(\mathbb{X}_m, \mathbb{Y}_n) := \mathbb{1}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n) > \widehat{q}^{\,\lambda}_{1-u_\alpha w_\lambda} \text{ for some } \lambda \in \Lambda\right)$$

positive weights $(w_\lambda)_{\lambda\in\Lambda}$ satisfying $\sum_{\lambda\in\Lambda} w_\lambda \leq 1$

Correction u_α defined as

$$\sup\left\{u > 0 : \mathbb{P}_{p\times p}\left(\max_{\lambda\in\Lambda}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n) - \widehat{q}^{\,\lambda}_{1-uw_\lambda}\right) > 0\right) \leq \alpha\right\}$$

more powerful than Bonferroni correction as $u_\alpha \geq \alpha$

Time complexity $\mathcal{O}\left(|\Lambda|\,(B_1 + B_2)\,(m+n)^2\right)$

MMDAgg for a *collection* of bandwidths Λ

MMDAgg (MMD Aggregation): non-asymptotic level α

$$\Delta_\alpha^\Lambda(\mathbb{X}_m, \mathbb{Y}_n) := \mathbb{1}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n) > \widehat{q}^{\,\lambda}_{1-u_\alpha w_\lambda} \text{ for some } \lambda \in \Lambda\right)$$

positive weights $(w_\lambda)_{\lambda\in\Lambda}$ satisfying $\sum_{\lambda\in\Lambda} w_\lambda \leq 1$

Correction u_α defined as

$$\sup\left\{u > 0 : \mathbb{P}_{p\times p}\left(\max_{\lambda\in\Lambda}\left(\widehat{\mathrm{MMD}}_\lambda^2(\mathbb{X}_m, \mathbb{Y}_n) - \widehat{q}^{\,\lambda}_{1-uw_\lambda}\right) > 0\right) \leq \alpha\right\}$$

more powerful than Bonferroni correction as $u_\alpha \geq \alpha$

Time complexity $\mathcal{O}\left(|\Lambda|\,(B_1 + B_2)\,(m+n)^2\right)$

MMDAgg for a *collection* of bandwidths Λ

MMDAgg (MMD Aggregation): non-asymptotic level α

$$\Delta_\alpha^\Lambda(\mathbb{X}_m, \mathbb{Y}_n) := \mathbb{1}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n) > \widehat{q}^{\,\lambda}_{1-u_\alpha w_\lambda} \text{ for some } \lambda \in \Lambda\right)$$

positive weights $(w_\lambda)_{\lambda\in\Lambda}$ satisfying $\sum_{\lambda\in\Lambda} w_\lambda \leq 1$

Correction u_α defined as

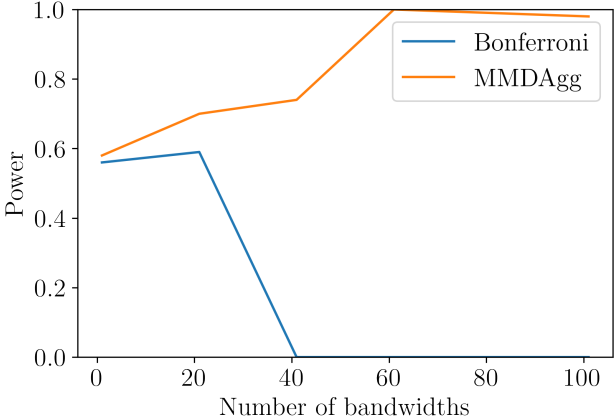
$$\sup\left\{u > 0 : \mathbb{P}_{p\times p}\left(\max_{\lambda\in\Lambda}\left(\widehat{\mathrm{MMD}}^2_\lambda(\mathbb{X}_m, \mathbb{Y}_n) - \widehat{q}^{\,\lambda}_{1-uw_\lambda}\right) > 0\right) \leq \alpha\right\}$$

more powerful than Bonferroni correction as $u_\alpha \geq \alpha$

Time complexity $\mathcal{O}\left(|\Lambda|\,(B_1 + B_2)\,(m+n)^2\right)$

Multiple testing correction comparison

Simple example: 3-d Gaussians with different means



$$\Lambda(i) := \left\{2^{\ell}\lambda_{\text{med}} : \ell \in \{-i, \dots, i\}\right\} \text{ for } i \in \{0, 10, 20, 30, 40, 50\}$$

$$w_\lambda := 1 \,/\, |\Lambda|$$

MMDAgg test power guarantee

Theorem (MMDAgg minimax adaptivity)

$$\Lambda^\star := \left\{ 2^{-\ell} \mathbb{1}_d : \ell \in \left\{ 1, \ldots, \left\lceil \frac{2}{d} \log_2 \left(\frac{m+n}{\ln(\ln(m+n))} \right) \right\rceil \right\} \right\}, \quad w_\lambda := \frac{6}{\pi^2 \, \ell^2}$$

Assuming $p - q \in \mathcal{S}_d^s(R)$, *the condition*

$$\|p - q\|_2 \;\geq\; \frac{C}{\sqrt{\beta}} \left(\frac{m+n}{\ln(\ln(m+n))} \right)^{-2s/(4s+d)}$$

guarantees control of the type II error of MMDAgg

$$\mathbb{P}_{p \times q}\left(\Delta_\alpha^{\Lambda^\star}(\mathbb{X}_m, \mathbb{Y}_n) = 0 \right) \;\leq\; \beta.$$

Minimax rate over Sobolev balls: $(m+n)^{-2s/(4s+d)}$

Minimax adaptive over $\{\mathcal{S}_d^s(R) : s > 0, R > 0\}$

Unlike the MMD test $\Delta_\alpha^{\lambda^\star}$, MMDAgg $\Delta_\alpha^{\Lambda^\star}$ is independent of s

MMDAgg parameter-free user-friendly implementation

Radial basis function (RBF) kernel: $k_\lambda(x, y) := K\left(\left\|\frac{x-y}{\lambda}\right\|\right)$

Collection of bandwidths Λ: discretisation of the interval $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the (robust) minimum and maximum of

$$\left\{\|x-y\| : x \in \mathbb{X}_m, y \in \mathbb{Y}_n\right\}$$

Possible to aggregate several kernels each with multiple bandwidths

Uniform weights: $w_\lambda := 1 \,/\, |\Lambda|$

Number of permutations / wild bootstraps: $B_1 = B_2 = 2000$

JAX: runs on either CPU or GPU (significant speed improvements)

- JAX GPU runs 100 times faster than Numpy CPU

mmdagg package: github.com/antoninschrab/mmdagg

```
from mmdagg import mmdagg                # X shape (m, d)
output = mmdagg(X, Y) # 0 or 1           # Y shape (n, d)
```

MMDAgg parameter-free user-friendly implementation

Radial basis function (RBF) kernel: $k_\lambda(x, y) := K\left(\left\|\frac{x-y}{\lambda}\right\|\right)$

Collection of bandwidths Λ: discretisation of the interval $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the (robust) minimum and maximum of

$$\left\{\|x-y\| : x \in \mathbb{X}_m, y \in \mathbb{Y}_n\right\}$$

Possible to aggregate several kernels each with multiple bandwidths

Uniform weights: $w_\lambda := 1 \,/\, |\Lambda|$

Number of permutations / wild bootstraps: $B_1 = B_2 = 2000$

JAX: runs on either CPU or GPU (significant speed improvements)

- JAX GPU runs 100 times faster than Numpy CPU

mmdagg package: github.com/antoninschrab/mmdagg

```
from mmdagg import mmdagg              # X shape (m, d)
output = mmdagg(X, Y) # 0 or 1         # Y shape (n, d)
```

MMDAgg parameter-free user-friendly implementation

Radial basis function (RBF) kernel: $k_\lambda(x, y) := K\left(\left\|\frac{x-y}{\lambda}\right\|\right)$

Collection of bandwidths Λ: discretisation of the interval $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the (robust) minimum and maximum of

$$\left\{\|x-y\| : x \in \mathbb{X}_m, y \in \mathbb{Y}_n\right\}$$

Possible to aggregate several kernels each with multiple bandwidths

Uniform weights: $w_\lambda := 1 / |\Lambda|$

Number of permutations / wild bootstraps: $B_1 = B_2 = 2000$

JAX: runs on either CPU or GPU (significant speed improvements)

- JAX GPU runs 100 times faster than Numpy CPU

mmdagg package: github.com/antoninschrab/mmdagg

```
from mmdagg import mmdagg              # X shape (m, d)
output = mmdagg(X, Y) # 0 or 1         # Y shape (n, d)
```

MMDAgg parameter-free user-friendly implementation

Radial basis function (RBF) kernel: $k_\lambda(x, y) := K\left(\left\|\frac{x-y}{\lambda}\right\|\right)$

Collection of bandwidths Λ: discretisation of the interval $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the (robust) minimum and maximum of

$$\left\{\|x-y\| : x \in \mathbb{X}_m, y \in \mathbb{Y}_n\right\}$$

Possible to aggregate several kernels each with multiple bandwidths

Uniform weights: $w_\lambda := 1 \,/\, |\Lambda|$

Number of permutations / wild bootstraps: $B_1 = B_2 = 2000$

JAX: runs on either CPU or GPU (significant speed improvements)

- JAX GPU runs 100 times faster than Numpy CPU

mmdagg package: github.com/antoninschrab/mmdagg

```
from mmdagg import mmdagg              # X shape (m, d)
output = mmdagg(X, Y) # 0 or 1         # Y shape (n, d)
```

MMDAgg parameter-free user-friendly implementation

Radial basis function (RBF) kernel: $k_\lambda(x, y) := K\left(\left\|\frac{x-y}{\lambda}\right\|\right)$

Collection of bandwidths Λ: discretisation of the interval $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the (robust) minimum and maximum of

$$\left\{\|x - y\| : x \in \mathbb{X}_m, y \in \mathbb{Y}_n\right\}$$

Possible to aggregate several kernels each with multiple bandwidths

Uniform weights: $w_\lambda := 1 \,/\, |\Lambda|$

Number of permutations / wild bootstraps: $B_1 = B_2 = 2000$

JAX: runs on either CPU or GPU (significant speed improvements)

- JAX GPU runs 100 times faster than Numpy CPU

mmdagg package: github.com/antoninschrab/mmdagg

```
from mmdagg import mmdagg            # X shape (m, d)
output = mmdagg(X, Y) # 0 or 1       # Y shape (n, d)
```

MMDAgg parameter-free user-friendly implementation

Radial basis function (RBF) kernel: $k_\lambda(x, y) := K\left(\left\|\frac{x-y}{\lambda}\right\|\right)$

Collection of bandwidths Λ: discretisation of the interval $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the (robust) minimum and maximum of

$$\left\{\|x - y\| : x \in \mathbb{X}_m, y \in \mathbb{Y}_n\right\}$$

Possible to aggregate several kernels each with multiple bandwidths

Uniform weights: $w_\lambda := 1 \,/\, |\Lambda|$

Number of permutations / wild bootstraps: $B_1 = B_2 = 2000$

JAX: runs on either CPU or GPU (significant speed improvements)

- JAX GPU runs 100 times faster than Numpy CPU

mmdagg package: github.com/antoninschrab/mmdagg

```
from mmdagg import mmdagg                # X shape (m, d)
output = mmdagg(X, Y) # 0 or 1           # Y shape (n, d)
```

MMDAgg parameter-free user-friendly implementation

Radial basis function (RBF) kernel: $k_\lambda(x, y) := K\left(\left\|\frac{x-y}{\lambda}\right\|\right)$

Collection of bandwidths Λ: discretisation of the interval $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the (robust) minimum and maximum of

$$\left\{\|x-y\| : x \in \mathbb{X}_m, y \in \mathbb{Y}_n\right\}$$

Possible to aggregate several kernels each with multiple bandwidths

Uniform weights: $w_\lambda := 1 \,/\, |\Lambda|$

Number of permutations / wild bootstraps: $B_1 = B_2 = 2000$

JAX: runs on either CPU or GPU (significant speed improvements)

- JAX GPU runs 100 times faster than Numpy CPU

mmdagg package: github.com/antoninschrab/mmdagg

```
from mmdagg import mmdagg                    # X shape (m, d)
output = mmdagg(X, Y) # 0 or 1               # Y shape (n, d)
```

MMDAgg parameter-free user-friendly implementation

Radial basis function (RBF) kernel: $k_\lambda(x, y) := K\left(\left\|\frac{x-y}{\lambda}\right\|\right)$

Collection of bandwidths Λ: discretisation of the interval $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the (robust) minimum and maximum of

$$\left\{\|x-y\| : x \in \mathbb{X}_m, y \in \mathbb{Y}_n\right\}$$

Possible to aggregate several kernels each with multiple bandwidths

Uniform weights: $w_\lambda := 1 \,/\, |\Lambda|$

Number of permutations / wild bootstraps: $B_1 = B_2 = 2000$

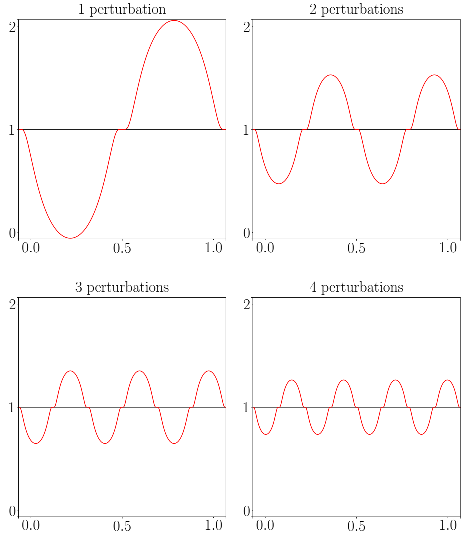
JAX: runs on either CPU or GPU (significant speed improvements)

- JAX GPU runs 100 times faster than Numpy CPU

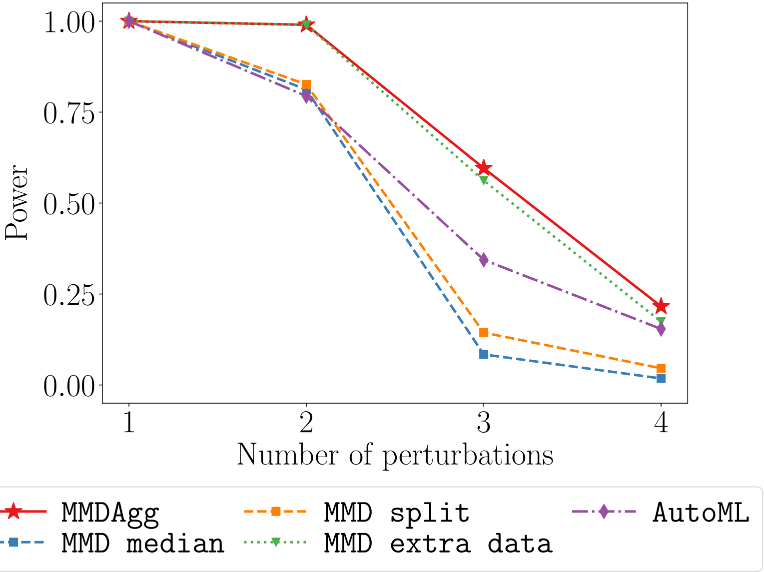
mmdagg package: github.com/antoninschrab/mmdagg

```
from mmdagg import mmdagg                 # X shape (m, d)
output = mmdagg(X, Y) # 0 or 1            # Y shape (n, d)
```

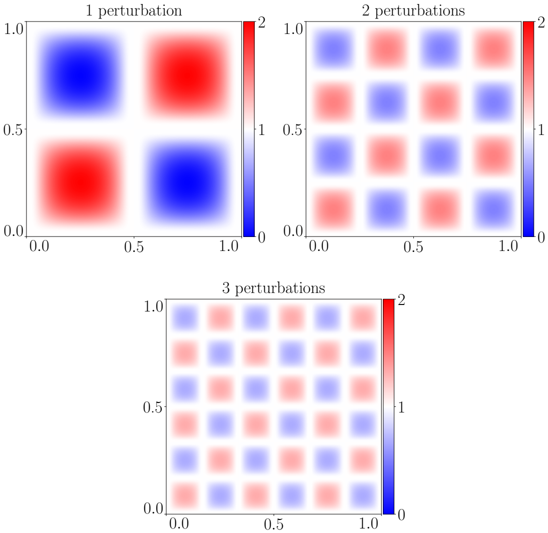
Experiment on perturbed uniform $d = 1$



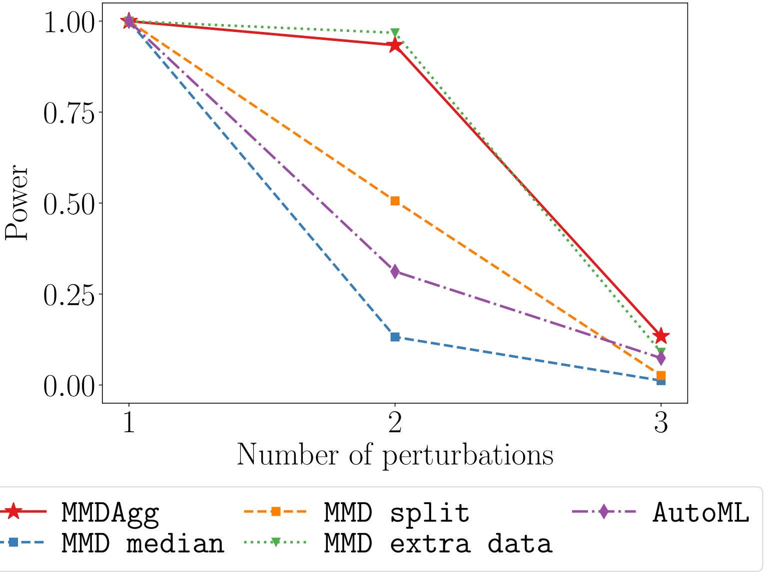
Experiment on perturbed uniform $d = 1$



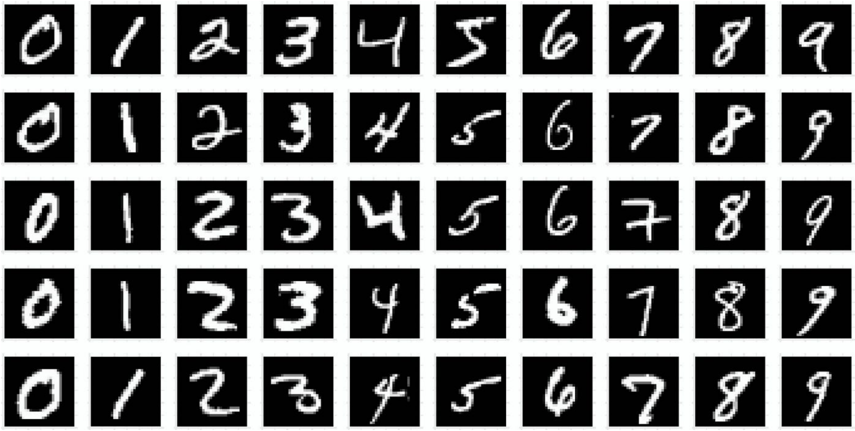
Experiment on perturbed uniform $d = 2$



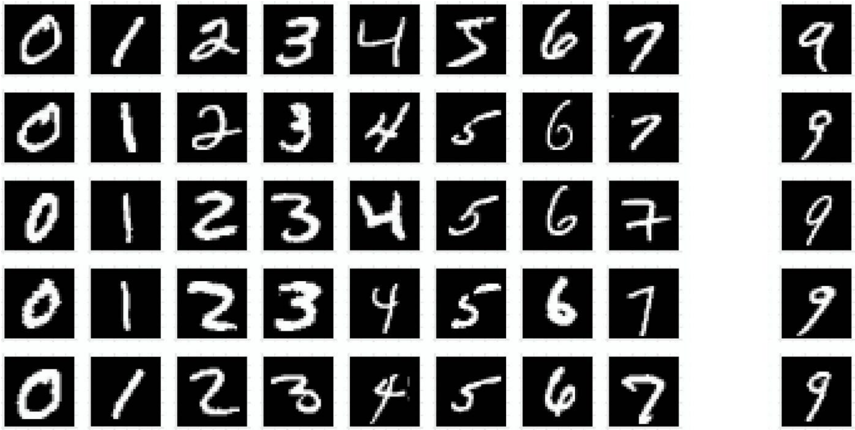
Experiment on perturbed uniform $d = 2$



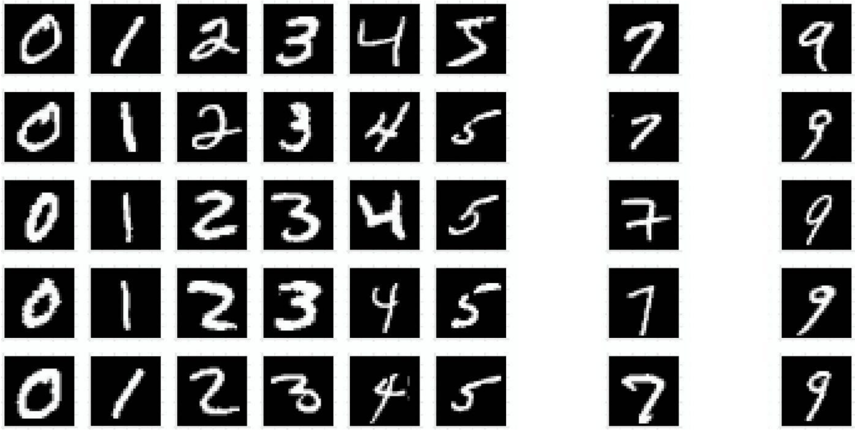
Experiment on MNIST digits



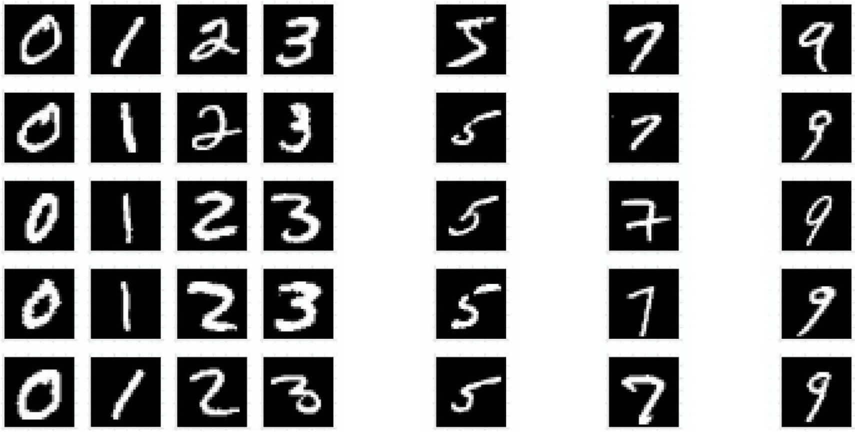
Experiment on MNIST digits



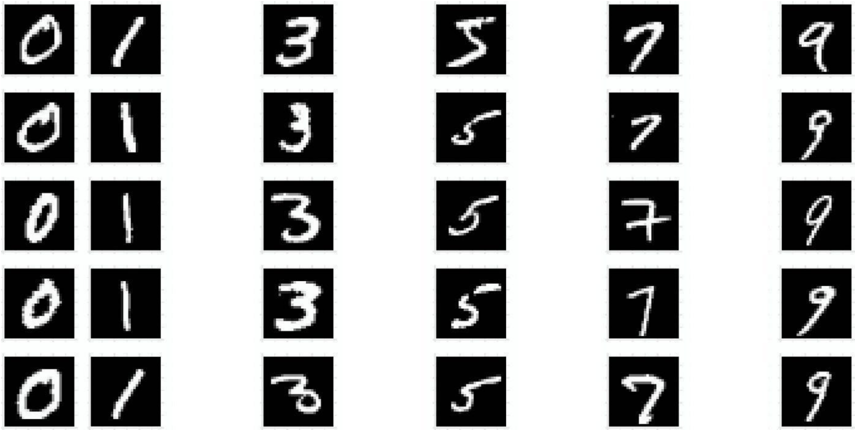
Experiment on MNIST digits



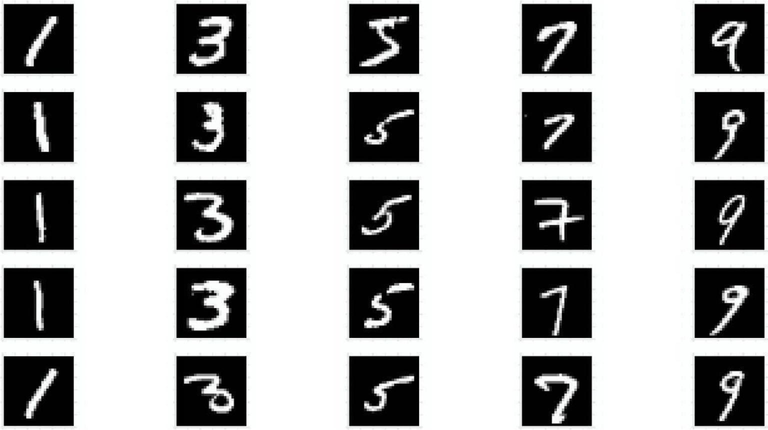
Experiment on MNIST digits



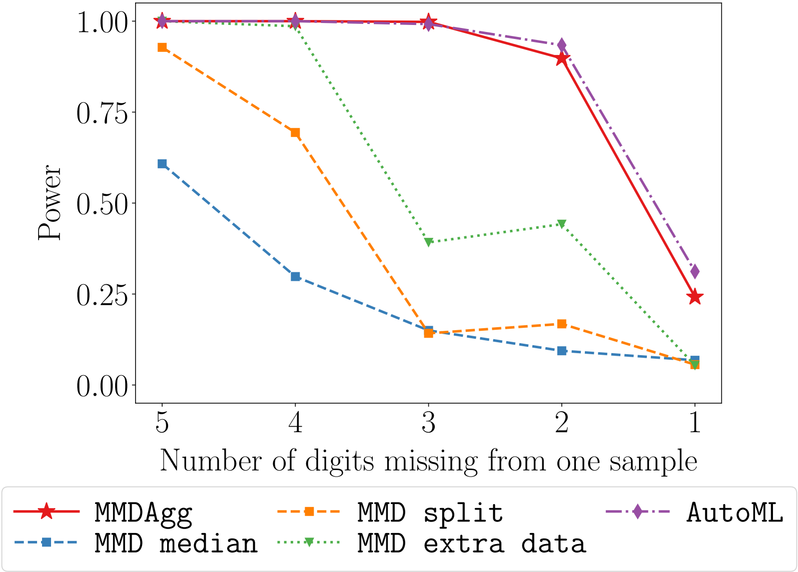
Experiment on MNIST digits



Experiment on MNIST digits

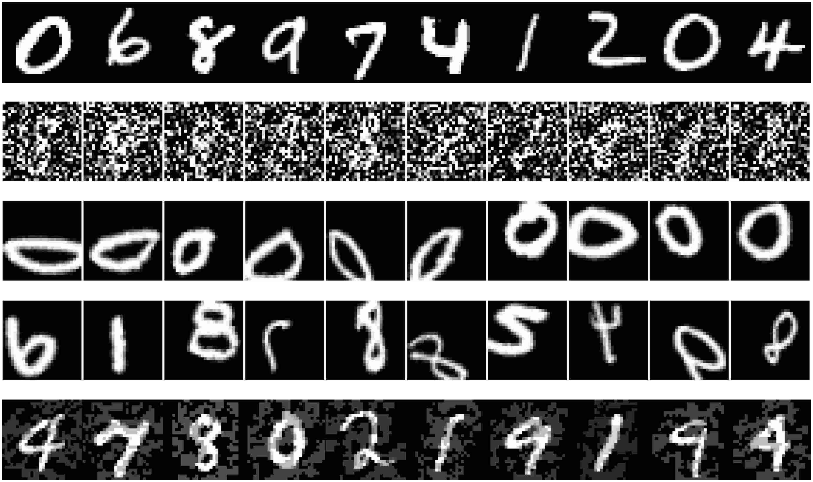


Experiment on MNIST digits



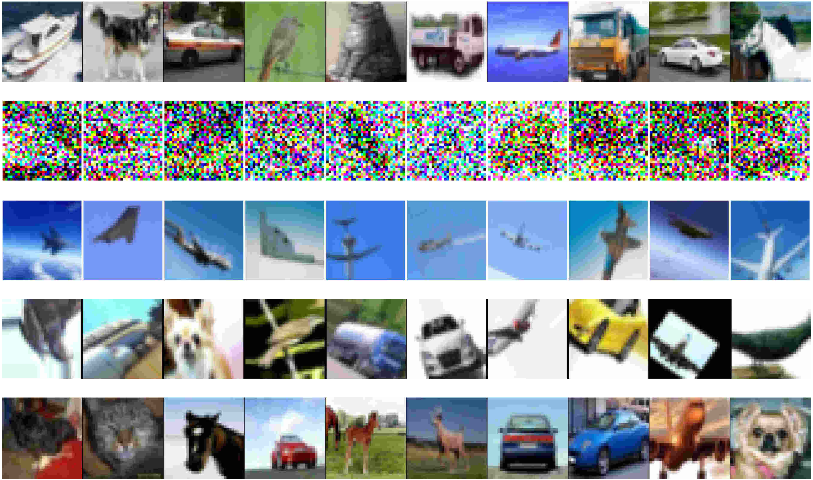
Experiment on image shifts on MNIST & CIFAR-10

Failing Loudly Benchmark: Rabanser et al., 2019

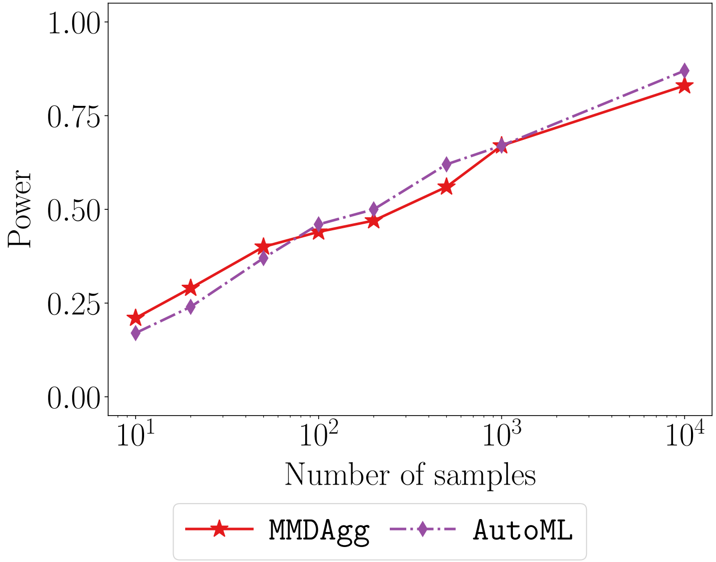


Experiment on image shifts on MNIST & CIFAR-10

Failing Loudly Benchmark: Rabanser et al., 2019



Experiment on image shifts on MNIST & CIFAR-10



MMD kernel choice without data splitting

MMD Aggregated Two-Sample Test (JMLR 2023):

arXiv > stat > arXiv:2110.15073

Statistics > Machine Learning

[Submitted on 28 Oct 2021 (v1), last revised 29 May 2023 (this version, v3)]

MMD Aggregated Two-Sample Test

Antonin Schrab, Ilmun Kim, Mélisande Albert, Béatrice Laurent, Benjamin Guedj, Arthur Gretton

Code:

https://github.com/antoninschrab/mmdagg-paper

Research support

Work supported by:

The Gatsby Charitable Foundation

Google Deepmind

Questions?


THE KERNEL BREWERY LONDON
BALTIC PORTER
THE KERNEL
EXPORT STOUT