Divergence measures for comparing distributions and training generative models: Part 1

Arthur Gretton

Gatsby Computational Neuroscience Unit,
University College London

DeepLearn, 2022

A motivation: comparing two samples

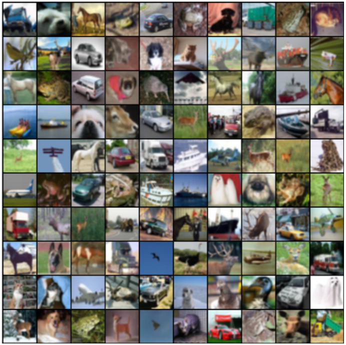
- Given: Samples from unknown distributions P and Q.
- Goal: do P and Q differ?

$$\sim P$$

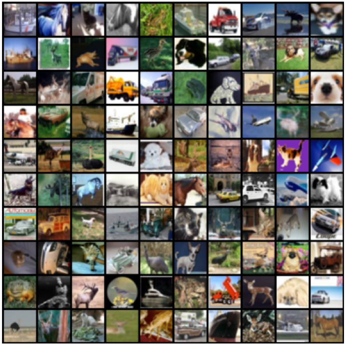
$$\sim Q$$

A real-life example: two-sample tests

- Goal: do P and Q differ?



CIFAR 10 samples



Cifar 10.1 samples

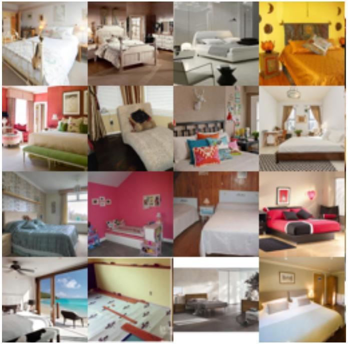
Significant difference?

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.

Training generative models

- Have: One collection of samples X from unknown distribution P.
- Goal: generate samples Q that look like P



LSUN bedroom samples P

Generated Q, MMD GAN

Training a Generative Adversarial Network

(Binkowski, Sutherland, Arbel, G., ICLR 2018),
(Arbel, Sutherland, Binkowski, G., NeurIPS 2018)

A second task: dependence testing

- Given: Samples from a distribution P_{XY}
- Goal: Are X and Y independent?

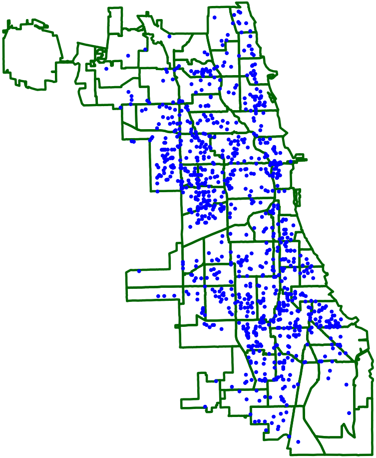
X	Y
	A large animal who slings slobber, exudes a distinctive houndy odor, and wants nothing more than to follow his nose.
	Their noses guide them through life, and they're never happier than when following an interesting scent.
	A responsive, interactive pet, one that will blow in your ear and follow you everywhere.

Text from dogtime.com and petfinder.com

A third task: testing goodness of fit

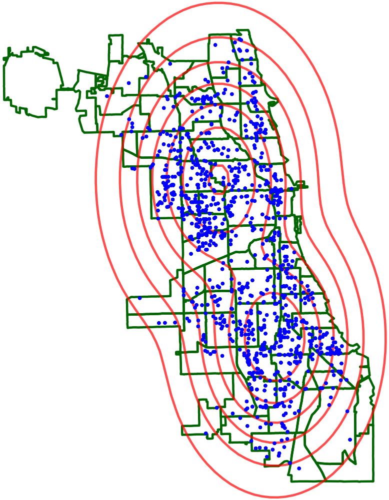
- Given: A model P and samples Q.
- Goal: is P a good fit for Q?

Chicago crime data



A third task: testing goodness of fit

- Given: A model P and samples Q.
- Goal: is P a good fit for Q?



Chicago crime data

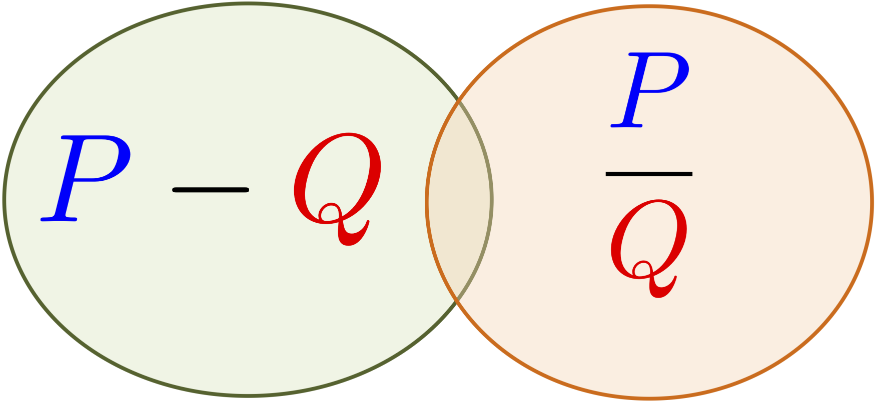
Model is Gaussian mixture with two components. Is this a good model?

Outline

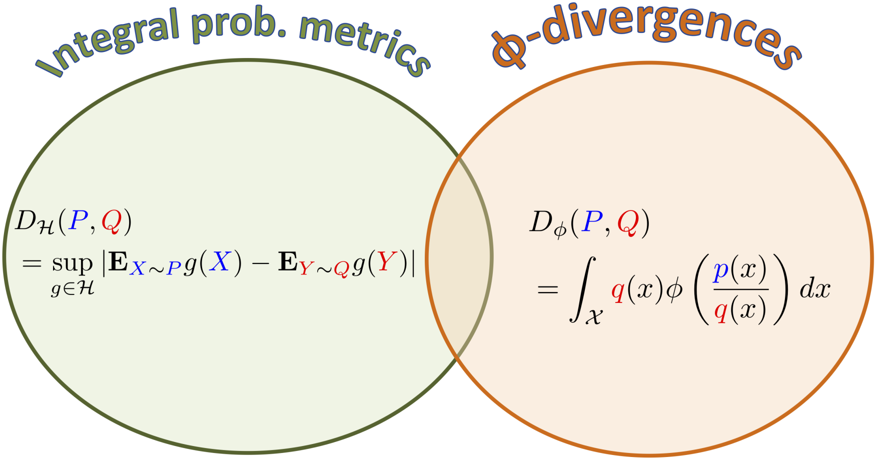
- Maximum Mean Discrepancy (MMD)...
  - ...as a difference in feature means
  - ...as an integral probability metric (not just a technicality!)
- A statistical test based on the MMD
  - learn adaptive NN features
- Training GANs generative adversarial networks with MMD
  - learn adaptive NN features
- Next parts:
  - ϕ-divergences for training GANS and Generalized Energy-Based models,
  - Kernel dependence measures, Stein discrepancies for goodness-of-fit (if time!)

Divergence measures

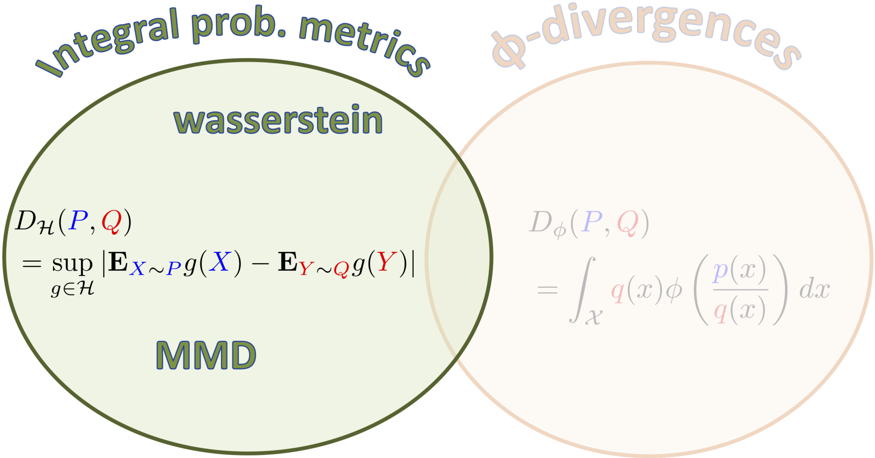
Divergences



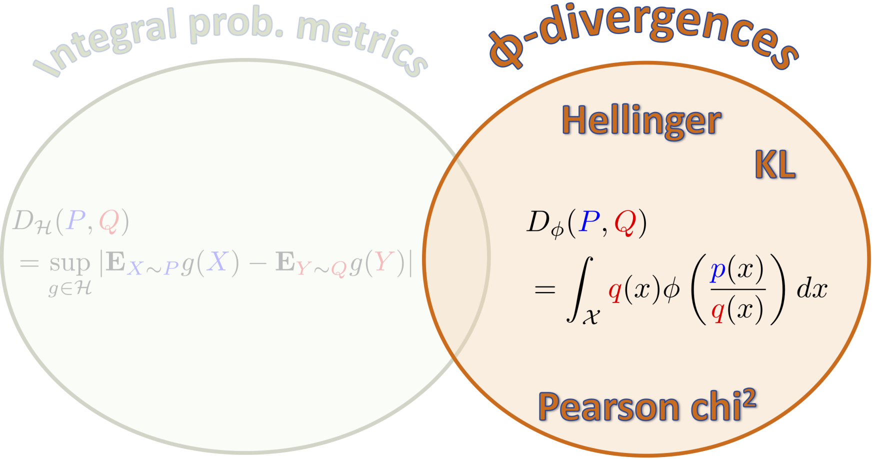
Divergences



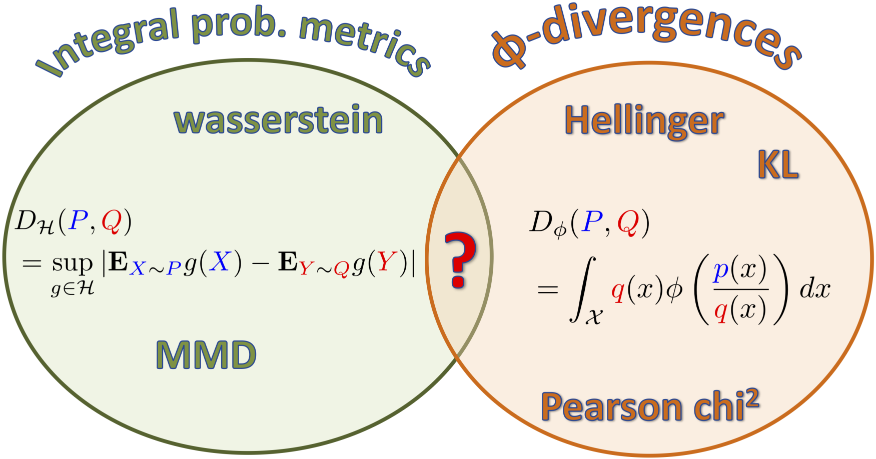
The integral probability metrics



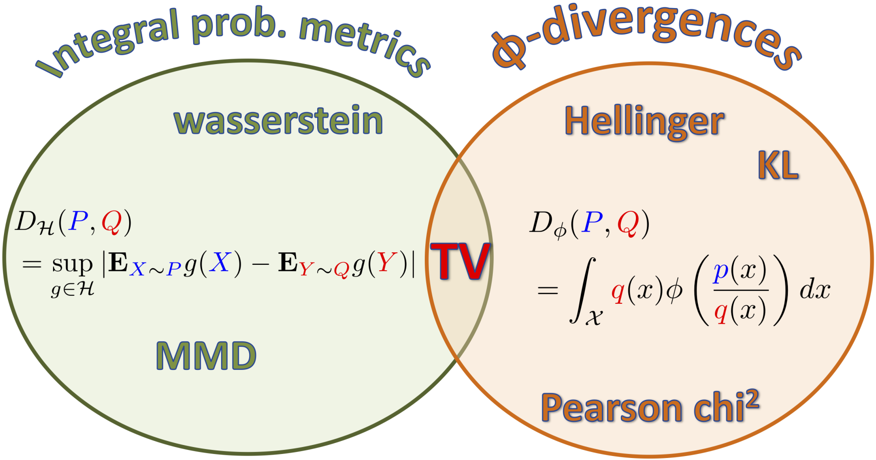
The ϕ-divergences



Divergences



Divergences

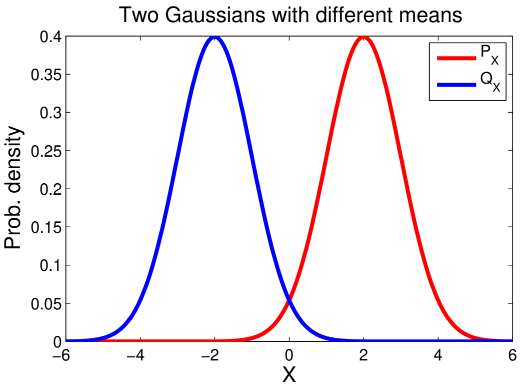


Sriperumbudur, Fukumizu, G, Schoelkopf, Lanckriet, EJS (2012)

The MMD

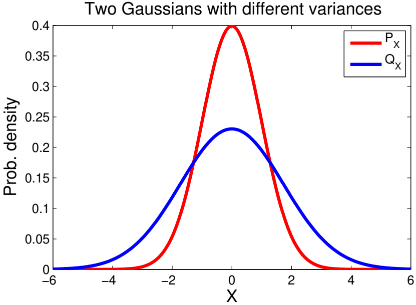
Feature mean difference

- Simple example: 2 Gaussians with different means
- Answer: t-test



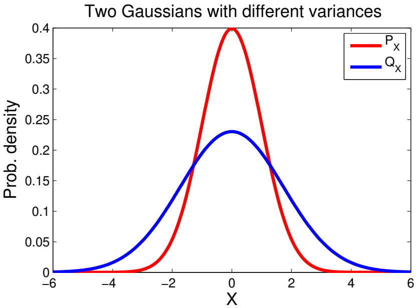
Feature mean difference

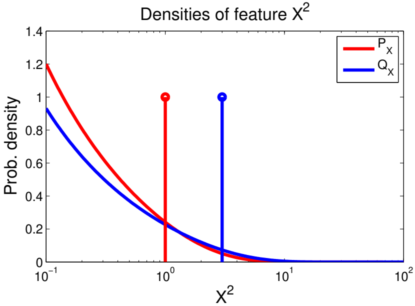
- Two Gaussians with same means, different variance
- Idea: look at difference in means of features of the RVs
- In Gaussian case: second order features of form $\varphi(x) = x^2$



Feature mean difference

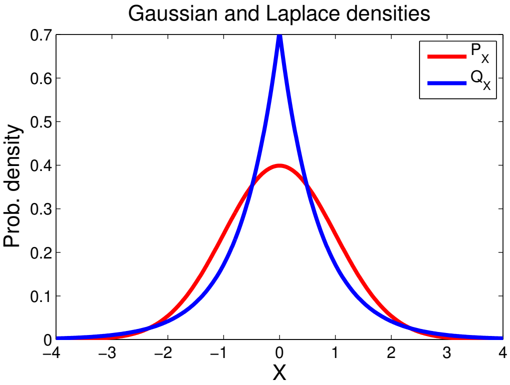
- Two Gaussians with same means, different variance
- Idea: look at difference in means of features of the RVs
- In Gaussian case: second order features of form $\varphi(x) = x^2$





Feature mean difference

- Gaussian and Laplace distributions
- Same mean *and* same variance
- Difference in means using higher order features...RKHS



Infinitely many features using kernels

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,

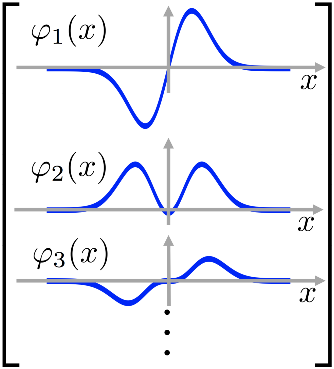
$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{F}}$$

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.

Infinitely many features of *distributions*

Given P a Borel probability measure on $\mathcal{X}$, define feature map of probability P,

$$\mu_P = [\ldots \mathrm{E}_P [\varphi_i(X)] \ldots]$$

For positive definite $k(x, x')$,

$$\langle \mu_P, \mu_Q \rangle_{\mathcal{F}} = \mathrm{E}_{P,Q} k(x, y)$$

for $x \sim P$ and $y \sim Q$.

Fine print: feature map $\varphi(x)$ must be Bochner integrable for all probability measures considered. Always true if kernel bounded.

Infinitely many features of *distributions*

Given P a Borel probability measure on $\mathcal{X}$, define feature map of probability P,

$$\mu_P = [\ldots \mathrm{E}_P [\varphi_i(X)] \ldots]$$

For positive definite $k(x, x')$,

$$\langle \mu_P, \mu_Q \rangle_{\mathcal{F}} = \mathrm{E}_{P,Q} k(x, y)$$

for $x \sim P$ and $y \sim Q$.

Fine print: feature map $\varphi(x)$ must be Bochner integrable for all probability measures considered. Always true if kernel bounded.

The maximum mean discrepancy

The maximum mean discrepancy is the distance between feature means:

$$
\begin{aligned}
MMD^2(P, Q) &= \|\mu_P - \mu_Q\|^2_{\mathcal{F}} \\
&= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{F}}
\end{aligned}
$$

The maximum mean discrepancy

The maximum mean discrepancy is the distance between feature means:

$$
\begin{aligned}
MMD^2(P, Q) &= \|\mu_P - \mu_Q\|^2_{\mathcal{F}} \\
&= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{F}} \\
&= \langle \mu_P, \mu_P \rangle_{\mathcal{F}} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{F}} - 2 \langle \mu_P, \mu_Q \rangle_{\mathcal{F}}
\end{aligned}
$$

The maximum mean discrepancy

The maximum mean discrepancy is the distance between feature means:

$$
\begin{aligned}
MMD^2(P, Q) &= \|\mu_P - \mu_Q\|^2_{\mathcal{F}} \\
&= \langle \mu_P - \mu_Q, \mu_P - \mu_Q \rangle_{\mathcal{F}} \\
&= \underbrace{\mathbf{E}_P k(X, X')}_{\text{(a)}} + \underbrace{\mathbf{E}_Q k(Y, Y')}_{\text{(a)}} - 2\underbrace{\mathbf{E}_{P,Q} k(X, Y)}_{\text{(b)}}
\end{aligned}
$$

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

Illustration of MMD

- Dogs ($= P$) and fish ($= Q$) example revisited
- Each entry is one of $k(\text{dog}_i, \text{dog}_j)$, $k(\text{dog}_i, \text{fish}_j)$, or $k(\text{fish}_i, \text{fish}_j)$

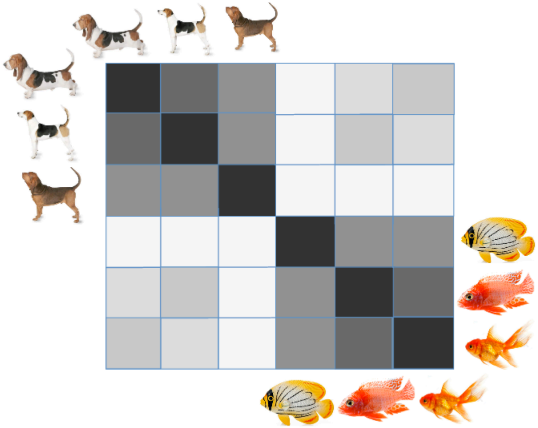
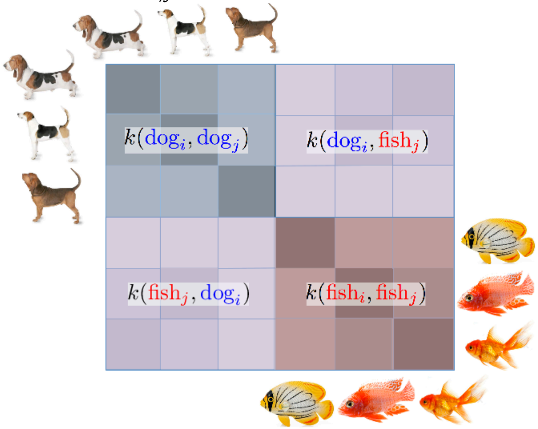


Illustration of MMD

The maximum mean discrepancy:

$$\widehat{MMD}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(\text{dog}_i, \text{dog}_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\text{fish}_i, \text{fish}_j) - \frac{2}{n^2} \sum_{i,j} k(\text{dog}_i, \text{fish}_j)$$

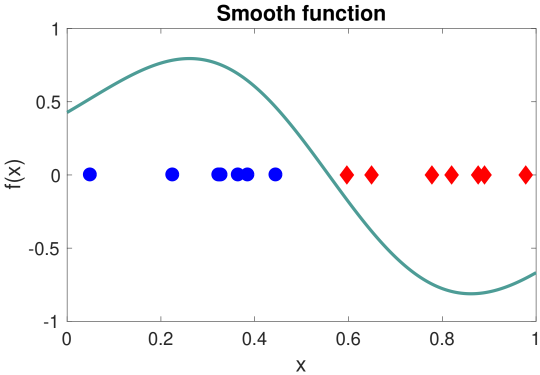


MMD as an integral probability metric

Integral probability metric:

Find a "well behaved function" $f(x)$ to maximize

$$\mathbf{E}_P f(X) - \mathbf{E}_Q f(Y)$$

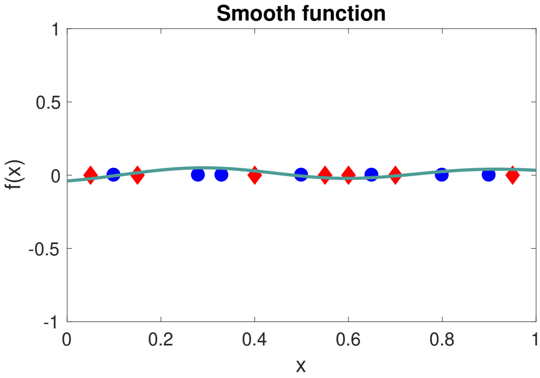


MMD as an integral probability metric

Integral probability metric:

Find a "well behaved function" $f(x)$ to maximize

$$E_P f(X) - E_Q f(Y)$$



MMD as an integral probability metric

Maximum mean discrepancy: smooth function for P vs Q

$$MMD(P, Q; F) := \sup_{\|f\|_{\mathcal{F}} \leq 1} \left[\mathbf{E}_P f(X) - \mathbf{E}_Q f(Y) \right]$$

$$(F = \text{unit ball in RKHS } \mathcal{F})$$

MMD as an integral probability metric

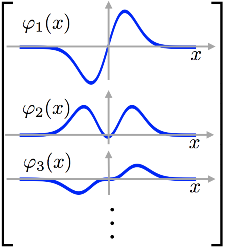
Maximum mean discrepancy: smooth function for P vs Q

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

(F = unit ball in RKHS $\mathcal{F}$)

Functions are linear combinations of features:

$$f(x) = \langle f, \varphi(x) \rangle_{\mathcal{F}} = \sum_{\ell=1}^{\infty} f_\ell \varphi_\ell(x) = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \end{bmatrix}^\top$$



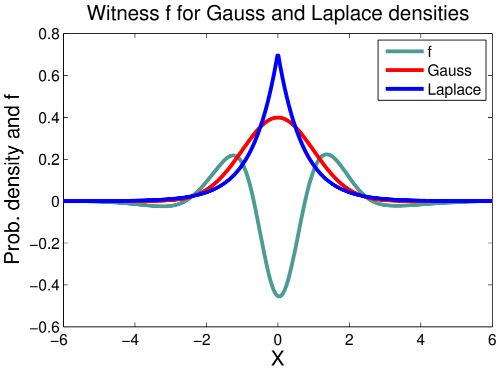
$$\|f\|_{\mathcal{F}}^2 := \sum_{i=1}^{\infty} f_i^2 \leq 1$$

MMD as an integral probability metric

Maximum mean discrepancy: smooth function for P vs Q

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

$$(F = \text{unit ball in RKHS } \mathcal{F})$$



MMD as an integral probability metric

Maximum mean discrepancy: smooth function for P vs Q

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

$$(F = \text{unit ball in RKHS } \mathcal{F})$$

For characteristic RKHS $\mathcal{F}$, $MMD(P, Q; F) = 0$ iff $P = Q$

Other choices for witness function class:

- Bounded continuous [Dudley, 2002]
- Bounded varation 1 (Kolmogorov metric) [Müller, 1997]
- Bounded Lipschitz (Wasserstein distances) [Dudley, 2002]

MMD as an integral probability metric

Maximum mean discrepancy: smooth function for P vs Q

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

$$(F = \text{unit ball in RKHS } \mathcal{F})$$

Expectations of functions are linear combinations of expected features

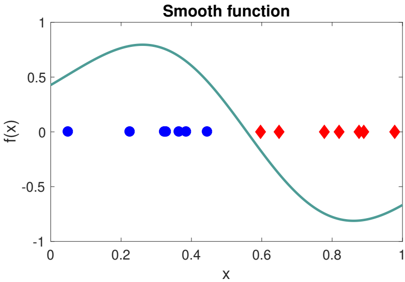
$$\mathrm{E}_P(f(X)) = \langle f, \mathrm{E}_P \varphi(X) \rangle_{\mathcal{F}} = \langle f, \mu_P \rangle_{\mathcal{F}}$$

(always true if kernel is bounded)

Integral prob. metric vs feature mean difference

The MMD:

$$MMD(P, Q; F) = \sup_{\|f\| \leq 1} \left[\mathrm{E}_P f(X) - \mathrm{E}_Q f(Y) \right]$$



Integral prob. metric vs feature mean difference

The MMD:

$$
\begin{aligned}
& MMD(P, Q; F) \\
&= \sup_{\|f\| \leq 1} [\mathrm{E}_P f(X) - \mathrm{E}_Q f(Y)] \\
&= \sup_{\|f\| \leq 1} \langle f, \mu_P - \mu_Q \rangle_{\mathcal{F}}
\end{aligned}
$$

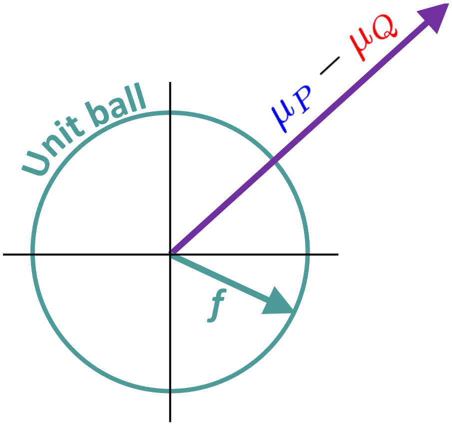
use

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

Integral prob. metric vs feature mean difference

The MMD:

$$MMD(P, Q; F)$$
$$= \sup_{\|f\|\leq 1} [\mathrm{E}_P f(X) - \mathrm{E}_Q f(Y)]$$
$$= \sup_{\|f\|\leq 1} \langle f, \mu_P - \mu_Q \rangle_{\mathcal{F}}$$



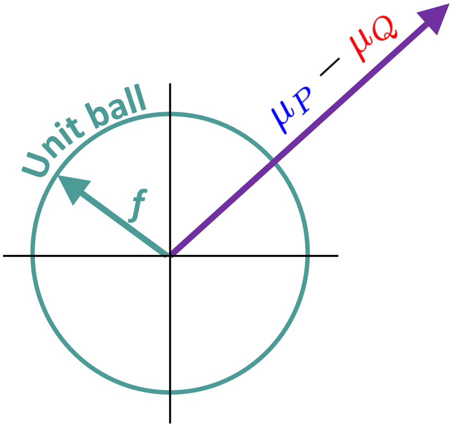
Integral prob. metric vs feature mean difference

The MMD:

$$MMD(P, Q; F)$$

$$= \sup_{\|f\| \leq 1} [\mathrm{E}_P f(X) - \mathrm{E}_Q f(Y)]$$

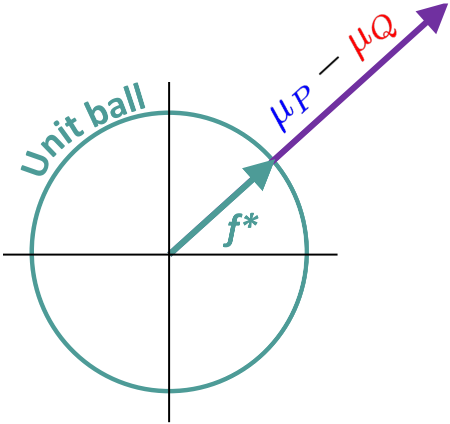
$$= \sup_{\|f\| \leq 1} \langle f, \mu_P - \mu_Q \rangle_{\mathcal{F}}$$



Integral prob. metric vs feature mean difference

The MMD:

$$MMD(P, Q; F) = \sup_{\|f\| \leq 1} [\mathrm{E}_P f(X) - \mathrm{E}_Q f(Y)] = \sup_{\|f\| \leq 1} \langle f, \mu_P - \mu_Q \rangle_{\mathcal{F}}$$



$$f^* = \frac{\mu_P - \mu_Q}{\|\mu_P - \mu_Q\|}$$

Integral prob. metric vs feature mean difference

The MMD:

$$
\begin{aligned}
&MMD(P, Q; F) \\
&= \sup_{\|f\|\leq 1} \left[\mathrm{E}_P f(X) - \mathrm{E}_Q f(Y) \right] \\
&= \sup_{\|f\|\leq 1} \langle f, \mu_P - \mu_Q \rangle_{\mathcal{F}} \\
&= \|\mu_P - \mu_Q\|_{\mathcal{F}}
\end{aligned}
$$

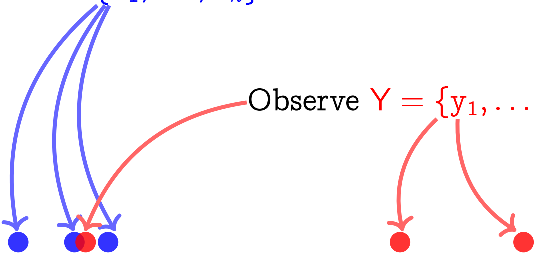
IPM view equivalent to feature mean difference (kernel case only)

Construction of MMD witness

Construction of empirical witness function (proof: next slide!)

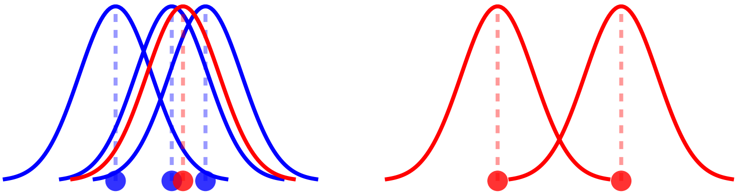
Observe $\mathsf{X} = \{\mathrm{x}_1, \ldots, \mathrm{x}_n\} \sim P$

Observe $\mathsf{Y} = \{\mathrm{y}_1, \ldots, \mathrm{y}_n\} \sim Q$



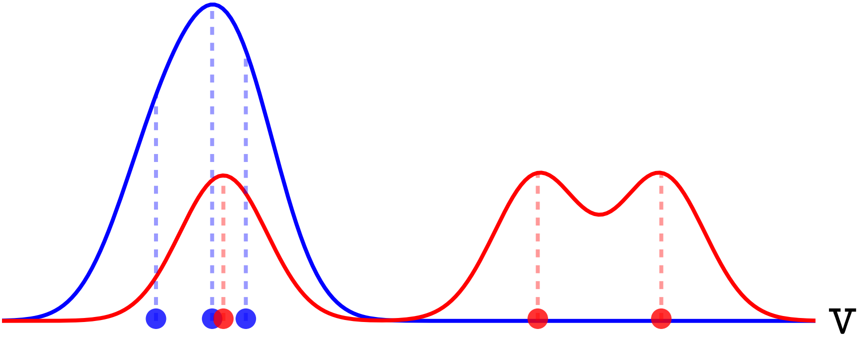
Construction of MMD witness

Construction of empirical witness function (proof: next slide!)



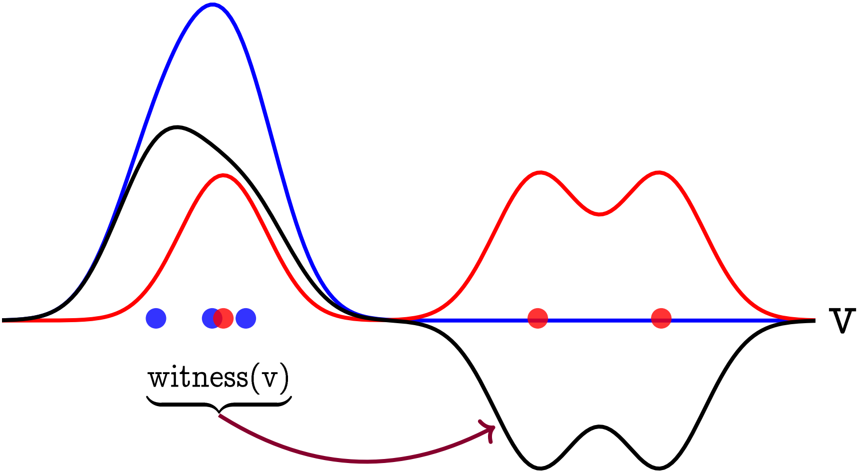
Construction of MMD witness

Construction of empirical witness function (proof: next slide!)



Construction of MMD witness

Construction of empirical witness function (proof: next slide!)



Derivation of empirical witness function

Recall the witness function expression

$$f^* \propto \mu_P - \mu_Q$$

Derivation of empirical witness function

Recall the witness function expression

$$f^* \propto \mu_P - \mu_Q$$

The empirical feature mean for P

$$\widehat{\mu}_P := \frac{1}{n} \sum_{i=1}^{n} \varphi(x_i)$$

Derivation of empirical witness function

Recall the witness function expression

$$f^* \propto \mu_P - \mu_Q$$

The empirical feature mean for P

$$\widehat{\mu}_P := \frac{1}{n}\sum_{i=1}^{n}\varphi(x_i)$$

The empirical witness function at v

$$f^*(v) = \langle f^*, \varphi(v)\rangle_{\mathcal{F}}$$

Derivation of empirical witness function

Recall the witness function expression

$$f^* \propto \mu_P - \mu_Q$$

The empirical feature mean for P

$$\widehat{\mu}_P := \frac{1}{n}\sum_{i=1}^{n}\varphi(x_i)$$

The empirical witness function at v

$$\begin{aligned} f^*(v) &= \langle f^*, \varphi(v)\rangle_{\mathcal{F}} \\ &\propto \langle \widehat{\mu}_P - \widehat{\mu}_Q, \varphi(v)\rangle_{\mathcal{F}} \end{aligned}$$

Derivation of empirical witness function

Recall the witness function expression

$$f^* \propto \mu_P - \mu_Q$$

The empirical feature mean for P

$$\widehat{\mu}_P := \frac{1}{n}\sum_{i=1}^{n}\varphi(x_i)$$

The empirical witness function at v

$$\begin{aligned} f^*(v) &= \langle f^*, \varphi(v)\rangle_{\mathcal{F}} \\ &\propto \langle \widehat{\mu}_P - \widehat{\mu}_Q, \varphi(v)\rangle_{\mathcal{F}} \\ &= \frac{1}{n}\sum_{i=1}^{n} k(x_i, v) - \frac{1}{n}\sum_{i=1}^{n} k(\mathrm{y}_i, v) \end{aligned}$$

Don't need explicit feature coefficients $f^* := \begin{bmatrix} f_1^* & f_2^* & \dots \end{bmatrix}$

Two-Sample Testing with MMD

A statistical test using MMD

The empirical MMD:

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

How does this help decide whether $P = Q$?

A statistical test using MMD

The empirical MMD:

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

Perspective from statistical hypothesis testing:

- Null hypothesis $\mathcal{H}_0$ when $P = Q$
  - should see $\widehat{MMD}^2$ "close to zero".
- Alternative hypothesis $\mathcal{H}_1$ when $P \neq Q$
  - should see $\widehat{MMD}^2$ "far from zero"

A statistical test using MMD

The empirical MMD:

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

Perspective from statistical hypothesis testing:

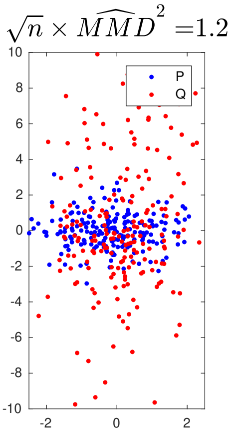
- Null hypothesis $\mathcal{H}_0$ when $P = Q$
  - should see $\widehat{MMD}^2$ "close to zero".
- Alternative hypothesis $\mathcal{H}_1$ when $P \neq Q$
  - should see $\widehat{MMD}^2$ "far from zero"

Want Threshold c_α for $\widehat{MMD}^2$ to get false positive rate α

Behaviour of $\widehat{MMD}^2$ when $P \neq Q$

Draw $n = 200$ i.i.d samples from P and Q

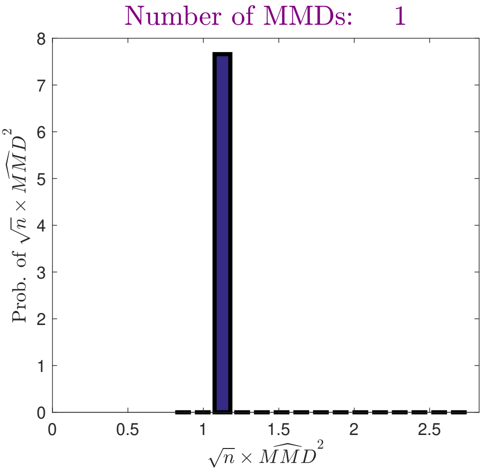
- Laplace with different y-variance.
- $\sqrt{n} \times \widehat{MMD}^2 = 1.2$

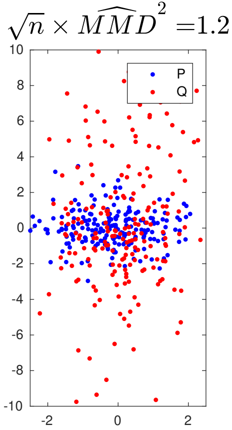


Behaviour of $\widehat{MMD}^2$ when $P \neq Q$

Draw $n = 200$ i.i.d samples from P and Q

- Laplace with different y-variance.
- $\sqrt{n} \times \widehat{MMD}^2 = 1.2$

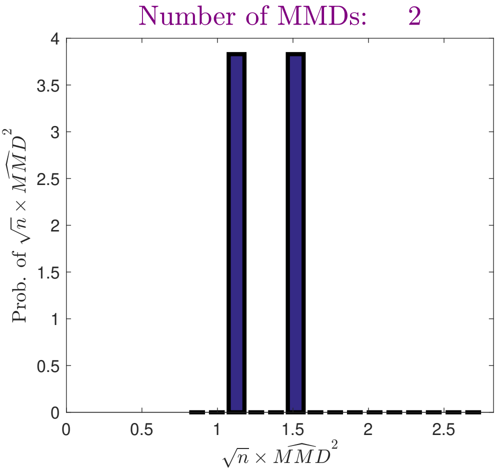


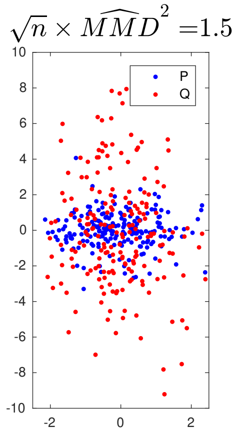


Behaviour of $\widehat{MMD}^2$ when $P \neq Q$

Draw $n = 200$ new samples from P and Q

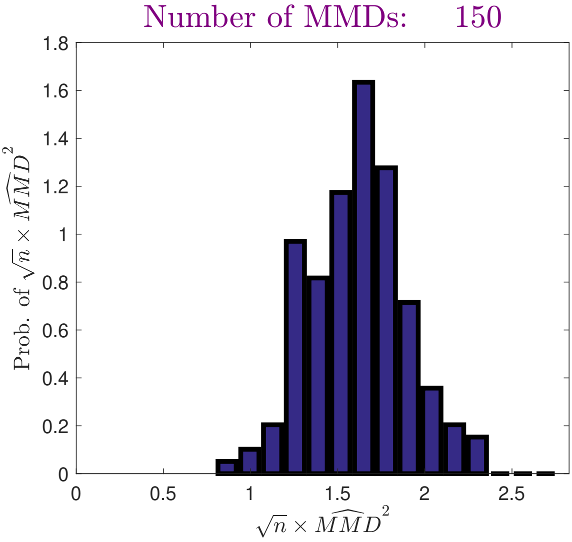
- Laplace with different y-variance.
- $\sqrt{n} \times \widehat{MMD}^2 = 1.5$





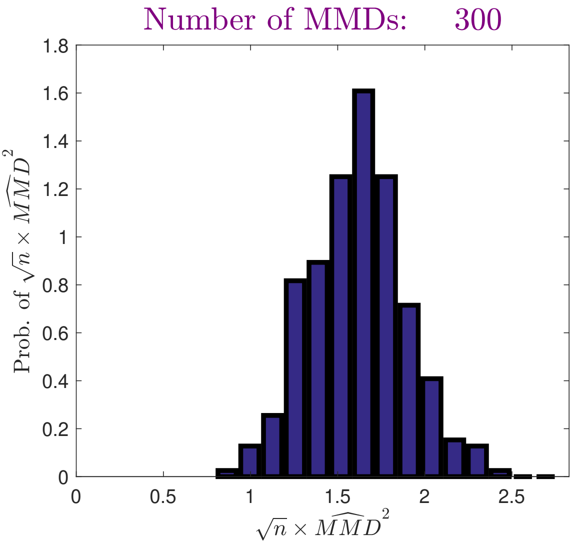
Behaviour of $\widehat{MMD}^2$ when $P \neq Q$

Repeat this 150 times . . .



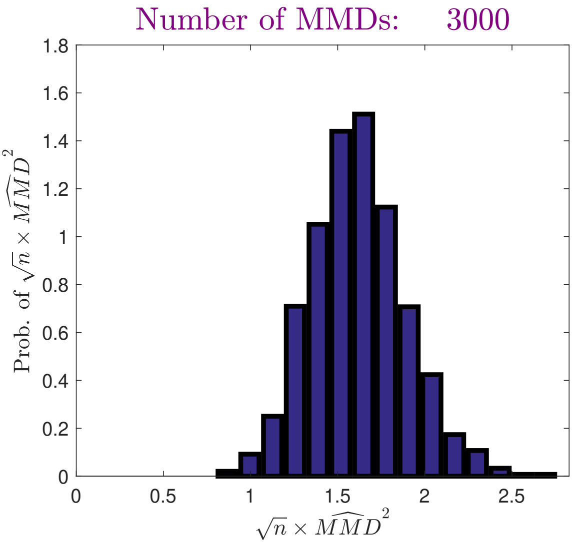
Behaviour of $\widehat{MMD}^2$ when $P \neq Q$

Repeat this 300 times . . .



Behaviour of $\widehat{MMD}^2$ when $P \neq Q$

Repeat this 3000 times ...

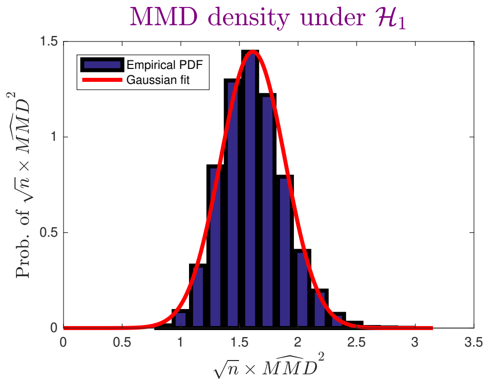


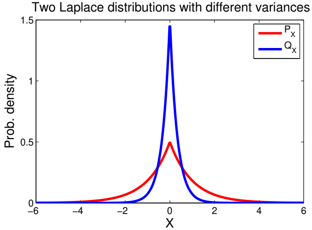
Asymptotics of $\widehat{MMD}^2$ when $P \neq Q$

When $P \neq Q$, statistic is asymptotically normal,

$$\frac{\widehat{\mathrm{MMD}}^2 - \mathrm{MMD}^2(P, Q)}{\sqrt{V_n(P, Q)}} \xrightarrow{D} \mathcal{N}(0, 1),$$

where variance $V_n(P, Q) = O\left(n^{-1}\right)$.



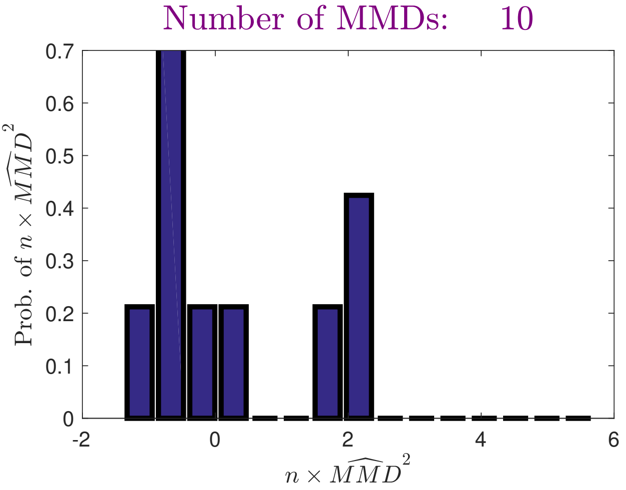


Behaviour of $\widehat{MMD}^2$ when $P = Q$

What happens when P and Q are the same?

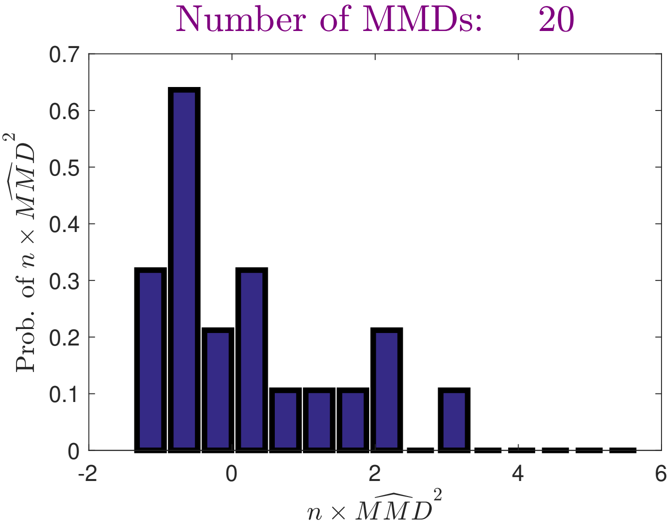
Behaviour of $\widehat{MMD}^2$ when $P = Q$

- Case of $P = Q = \mathcal{N}(0, 1)$



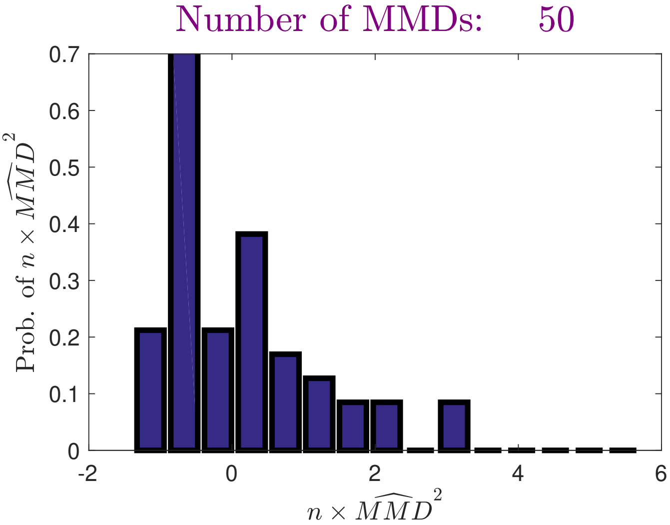
Behaviour of $\widehat{MMD}^2$ when $P = Q$

- Case of $P = Q = \mathcal{N}(0,1)$



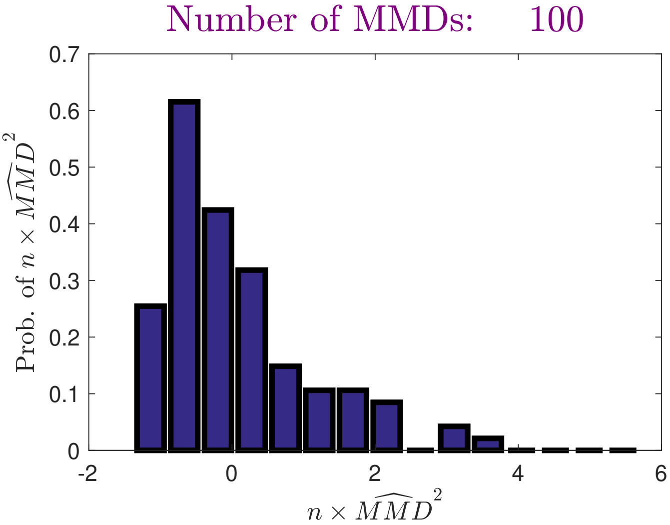
Behaviour of $\widehat{MMD}^2$ when $P = Q$

- Case of $P = Q = \mathcal{N}(0, 1)$



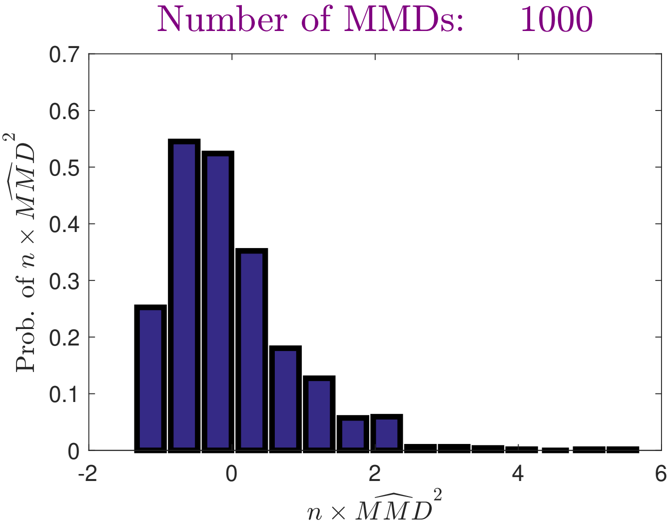
Behaviour of $\widehat{MMD}^2$ when $P = Q$

- Case of $P = Q = \mathcal{N}(0,1)$



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

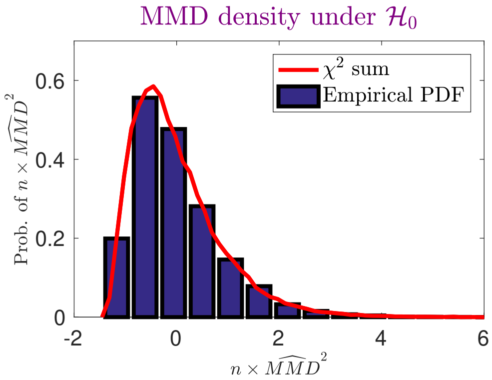
- Case of $P = Q = \mathcal{N}(0,1)$



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

Where $P = Q$, statistic has asymptotic distribution

$$n\widehat{\text{MMD}}^2 \sim \sum_{l=1}^{\infty} \lambda_l \left[z_l^2 - 2\right]$$



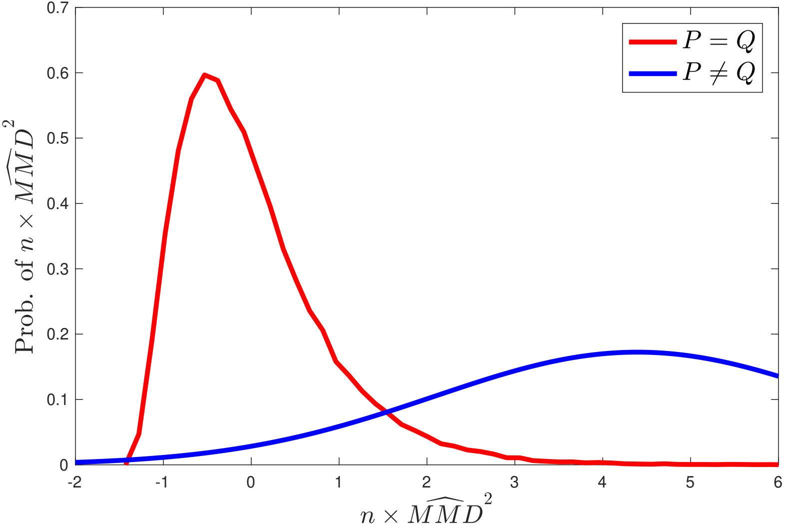
where

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

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

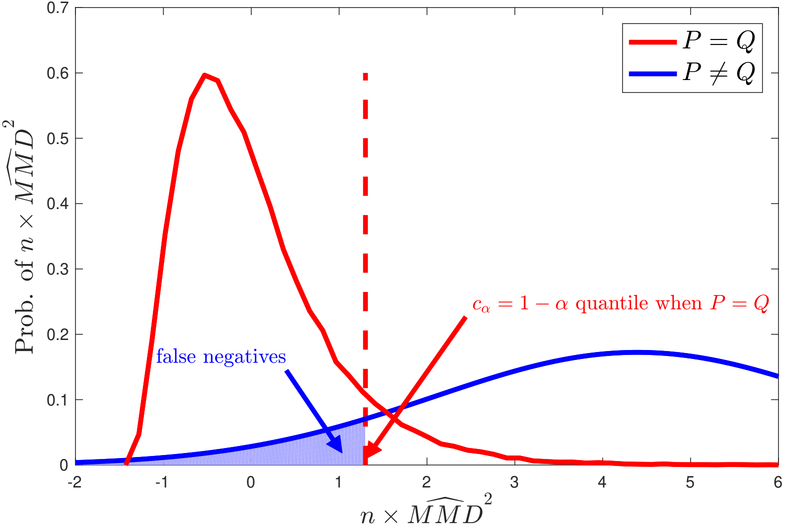
A statistical test

A summary of the asymptotics:



A statistical test

Test construction: (G., Borgwardt, Rasch, Schoelkopf, and Smola, JMLR 2012)



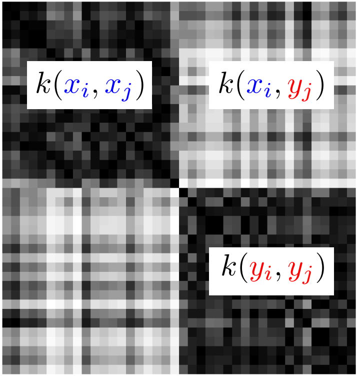
How do we get test threshold c_α?

Original empirical MMD for dogs and fish:

$X = [$  $\ldots\ ]$

$Y = [$  $\ldots\ ]$

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

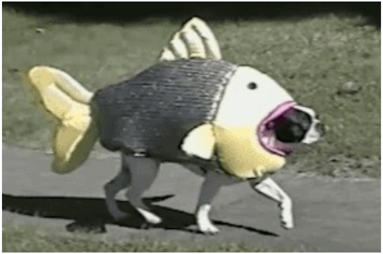


How do we get test threshold c_α?

Permuted dog and fish samples (merdogs):

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

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



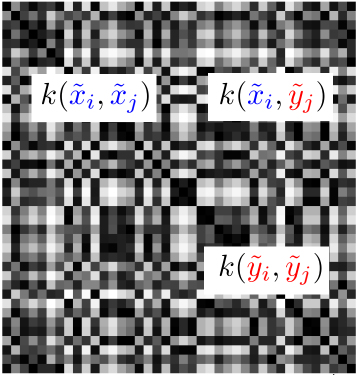
How do we get test threshold c_α?

Permuted dog and fish samples (merdogs):

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

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

$$\widehat{MMD}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(\tilde{x}_i, \tilde{x}_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\tilde{y}_i, \tilde{y}_j) - \frac{2}{n^2} \sum_{i,j} k(\tilde{x}_i, \tilde{y}_j)$$



Permutation simulates

$P = Q$

How do we get test threshold c_α?

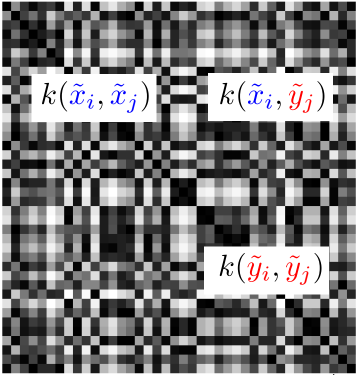
Permuted dog and fish samples (merdogs):

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

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

Exact level α (upper bound on false positive rate) at finite n and number of permutations (when unpermuted statistic included in pool)

Proposition 1, Schrab, Kim, Albert, Laurent, Guedj, Gretton (2021), MMD Aggregated Two-Sample Test, arXiv:2110.15073



How to choose the best kernel: optimising the kernel parameters

The best test for the job

- A test's power depends on $k(x, x')$, P, and Q (and n)
- With characteristic kernel, MMD test has power $\to 1$ as $n \to \infty$ for any (fixed) problem
  - But, for many P and Q, will have terrible power with reasonable n!

The best test for the job

- A test's power depends on $k(x, x')$, P, and Q (and n)
- With characteristic kernel, MMD test has power $\to 1$ as $n \to \infty$ for any (fixed) problem
  - But, for many P and Q, will have terrible power with reasonable n!
- You *can* choose a good kernel for a given problem
- You *can't* get one kernel that has good finite-sample power for all problems
  - "No one test can have all that power"

Choosing a kernel for the test

- Simple choice: exponentiated quadratic

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$$

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

Choosing a kernel for the test

- Simple choice: exponentiated quadratic

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$$

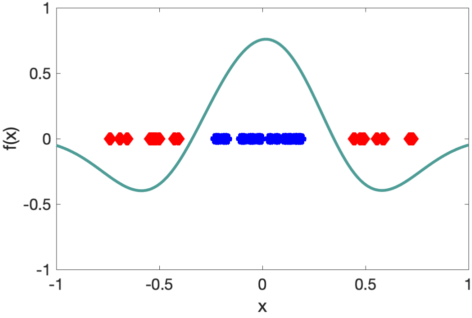
- *Characteristic:* for any σ: for any P and Q, power $\to 1$ as $n \to \infty$
- But choice of σ is very important for finite n...

Choosing a kernel for the test

- Simple choice: exponentiated quadratic

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$$

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

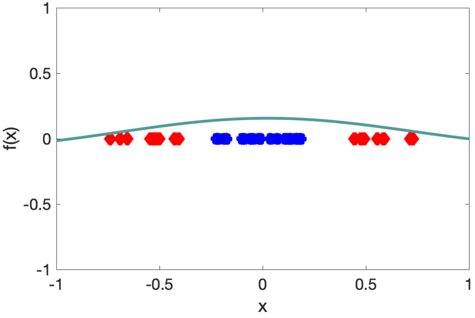


Choosing a kernel for the test

- Simple choice: exponentiated quadratic

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$$

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

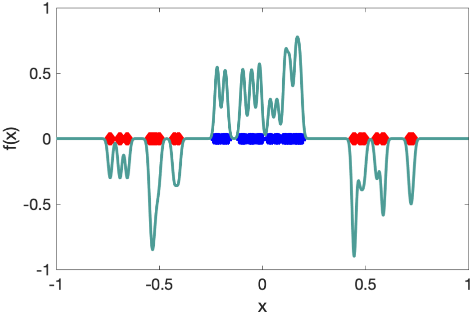


Choosing a kernel for the test

- Simple choice: exponentiated quadratic

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$$

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



Choosing a kernel for the test

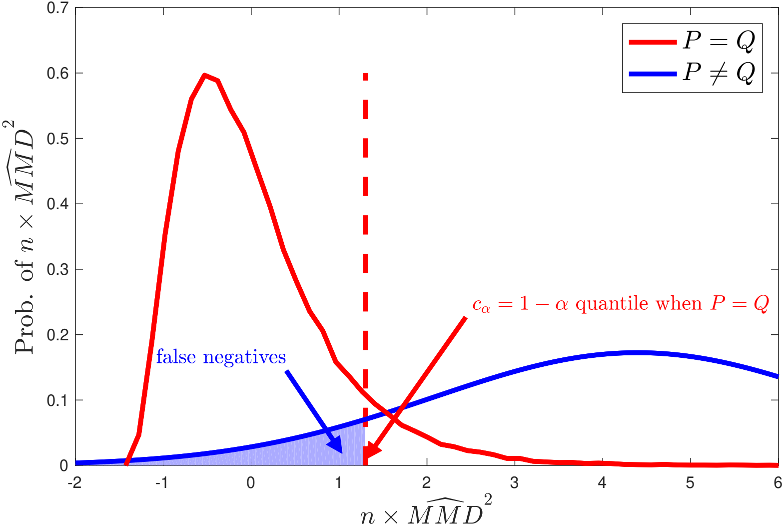
- Simple choice: exponentiated quadratic

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$$

- *Characteristic:* for any σ: for any P and Q, power $\to 1$ as $n \to \infty$
- But choice of σ is very important for finite n...
- ...and some problems (e.g. images) might have no good choice for σ

Graphical illustration

- Maximising test power same as minimizing false negatives



Optimizing kernel for test power

The power of our test ($\Pr_1$ denotes probability under $P \neq Q$):

$$\Pr_1 \left(n\widehat{\mathrm{MMD}}^2 > \hat{c}_\alpha \right)$$

Optimizing kernel for test power

The power of our test ($\Pr_1$ denotes probability under $P \neq Q$):

$$\Pr_1\left(n\widehat{\text{MMD}}^2 > \hat{c}_\alpha\right) \to \Phi\left(\frac{\text{MMD}^2(P,Q)}{\sqrt{V_n(P,Q)}} - \frac{c_\alpha}{n\sqrt{V_n(P,Q)}}\right)$$

where

- Φ is the CDF of the standard normal distribution.
- $\hat{c}_\alpha$ is an estimate of c_α test threshold.

Optimizing kernel for test power

The power of our test ($\Pr_1$ denotes probability under $P \neq Q$):

$$\Pr_1\left(n\widehat{\mathrm{MMD}}^2 > \hat{c}_\alpha\right)$$
$$\to \Phi\left(\underbrace{\frac{\mathrm{MMD}^2(P,Q)}{\sqrt{V_n(P,Q)}}}_{O(n^{1/2})} - \underbrace{\frac{c_\alpha}{n\sqrt{V_n(P,Q)}}}_{O(n^{-1/2})}\right)$$

For large n, second term negligible!

Optimizing kernel for test power

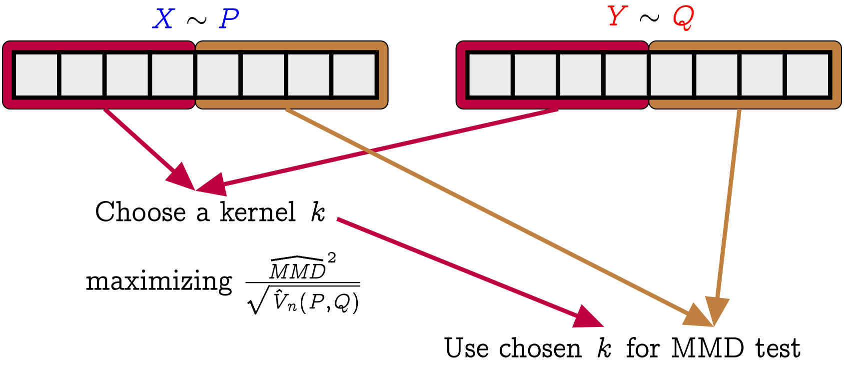
The power of our test ($\Pr_1$ denotes probability under $P \neq Q$):

$$\Pr_1\left(n\widehat{\mathrm{MMD}}^2 > \hat{c}_\alpha\right)$$
$$\to \Phi\left(\frac{\mathrm{MMD}^2(P,Q)}{\sqrt{V_n(P,Q)}} - \frac{c_\alpha}{n\sqrt{V_n(P,Q)}}\right)$$

To maximize test power, maximize

$$\frac{\mathrm{MMD}^2(P,Q)}{\sqrt{V_n(P,Q)}}$$

Data splitting

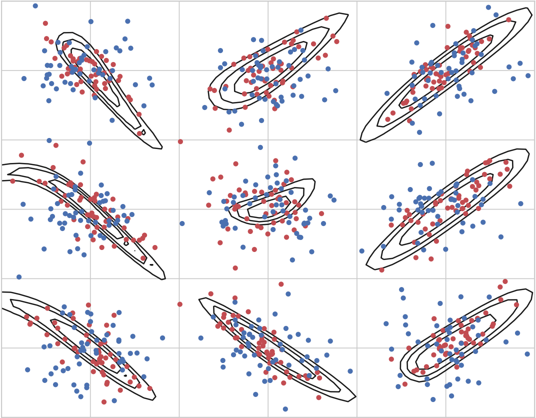


Learning a kernel helps a lot

Kernel with deep learned features:

$$k_\theta(x, y) = [(1 - \epsilon)\kappa(\Phi_\theta(x), \Phi_\theta(y)) + \epsilon]\, q(x, y)$$

κ and q are Gaussian kernels



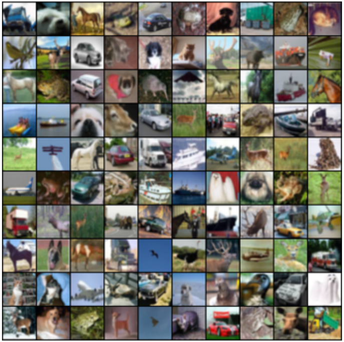
Learning a kernel helps a lot

Kernel with deep learned features:

$$k_\theta(x, y) = [(1 - \epsilon)\kappa(\Phi_\theta(x), \Phi_\theta(y)) + \epsilon]\, q(x, y)$$

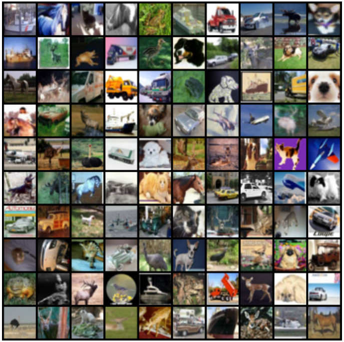
κ and q are Gaussian kernels

- CIFAR-10 vs CIFAR-10.1, null rejected 75% of time



CIFAR-10 test set (Krizhevsky 2009)

$X \sim P$



CIFAR-10.1 (Recht+ ICML 2019)

$Y \sim Q$

Learning a kernel helps a lot

Kernel with deep learned features:

$$k_\theta(x, y) = [(1 - \epsilon)\kappa(\Phi_\theta(x), \Phi_\theta(y)) + \epsilon]\, q(x, y)$$

κ and q are Gaussian kernels

- CIFAR-10 vs CIFAR-10.1, null rejected 75% of time

arXiv.org > stat > arXiv:2002.09116

Statistics > Machine Learning

[Submitted on 21 Feb 2020]

Learning Deep Kernels for Non-Parametric Two-Sample Tests

Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, D. J. Sutherland

ICML 2020

Code: https://github.com/fengliu90/DK-for-TST

Adaptive testing without data splitting?

Adaptive testing without data splitting?

arXiv > stat > arXiv:2110.15073

Statistics > Machine Learning

[Submitted on 28 Oct 2021]

MMD Aggregated Two-Sample Test

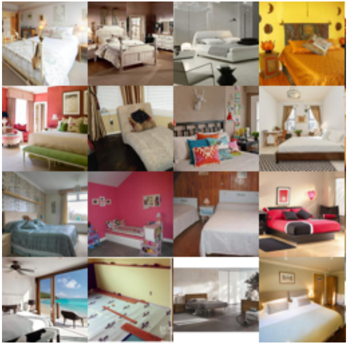
Antonin Schrab, Ilmun Kim, Mélisande Albert, Béatrice Laurent, Benjamin Guedj, Arthur Gretton

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

MMD for GAN training

Training implicit generative models

- Have: One collection of samples X from unknown distribution P.
- Goal: generate samples Q that look like P



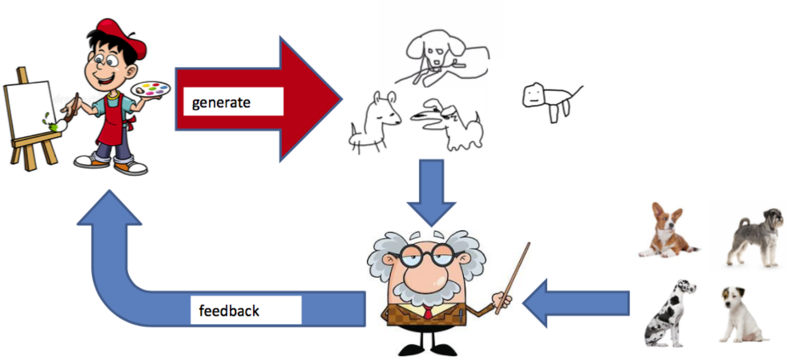
LSUN bedroom samples P

Generated Q, MMD GAN

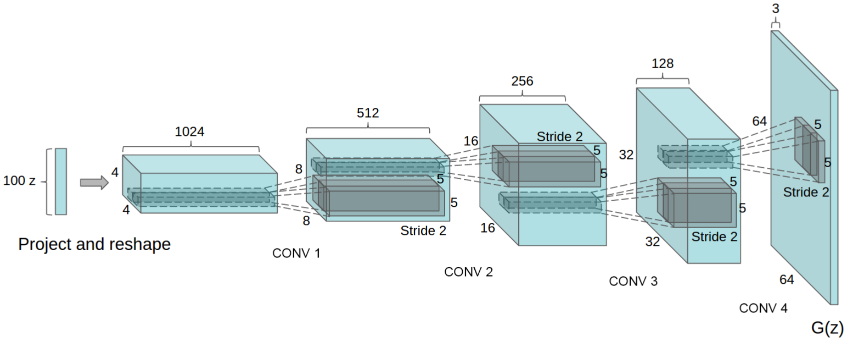
Using a critic $D(P, Q)$ to train a GAN

(Binkowski, Sutherland, Arbel, G., ICLR 2018),
(Arbel, Sutherland, Binkowski, G., NeurIPS 2018)

Visual notation: GAN setting



What I *won't* cover yet: the generator



Radford, Metz, Chintala, ICLR 2016

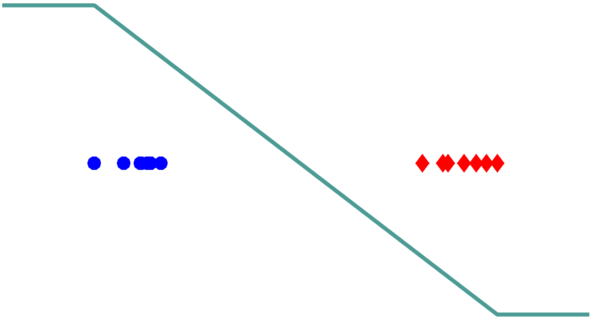
Wasserstein distance as critic

A helpful critic witness:

$$W_1(P, Q) = \sup_{\|f\|_L \leq 1} E_P f(X) - E_Q f(Y).$$

$$\|f\|_L := \sup_{x \neq y} |f(x) - f(y)| / \|x - y\|$$

$W_1 = 0.88$



Santambrogio, Optimal Transport for Applied Mathematicians (2015, Section 5.4)
G Peyré, M Cuturi, Computational Optimal Transport (2019)
M. Cuturi, J. Solomon, NeurIPS tutorial (2017)

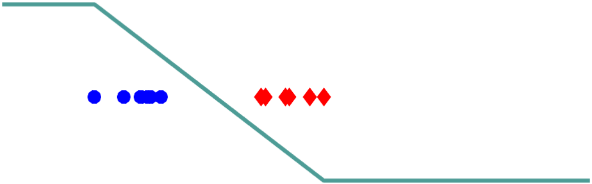
Wasserstein distance as critic

A helpful critic witness:

$$W_1(P, Q) = \sup_{\|f\|_L \le 1} E_P f(X) - E_Q f(Y).$$

$$\|f\|_L := \sup_{x \neq y} |f(x) - f(y)| / \|x - y\|$$

W_1=0.65



Santambrogio, Optimal Transport for Applied Mathematicians (2015, Section 5.4)

G Peyré, M Cuturi, Computational Optimal Transport (2019)

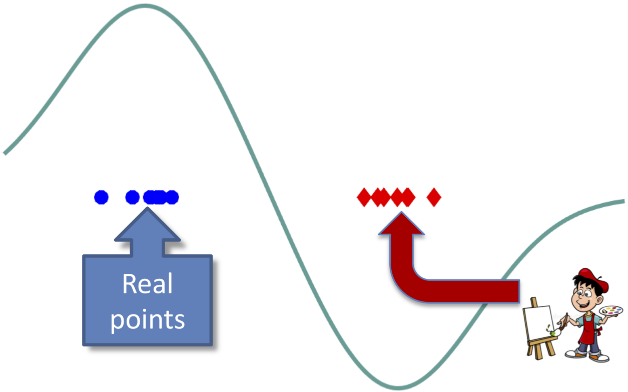
M. Cuturi, J. Solomon, NeurIPS tutorial (2017)

MMD as critic

A helpful critic witness:

$$MMD(P,Q) = \sup_{\|f\|_{\mathcal{F}}\leq 1} E_P f(X) - E_Q f(Y).$$

MMD=1.8

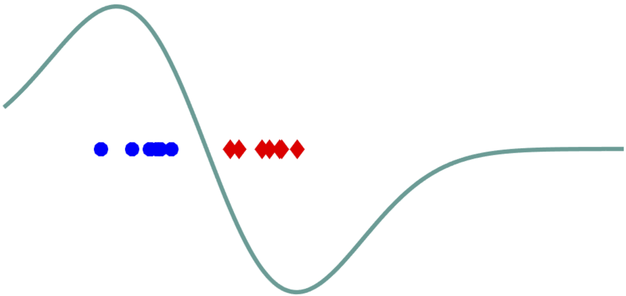


MMD as critic

A helpful critic witness:

$$MMD(P, Q) = \sup_{\|f\|_{\mathcal{F}} \leq 1} E_P f(X) - E_Q f(Y)$$

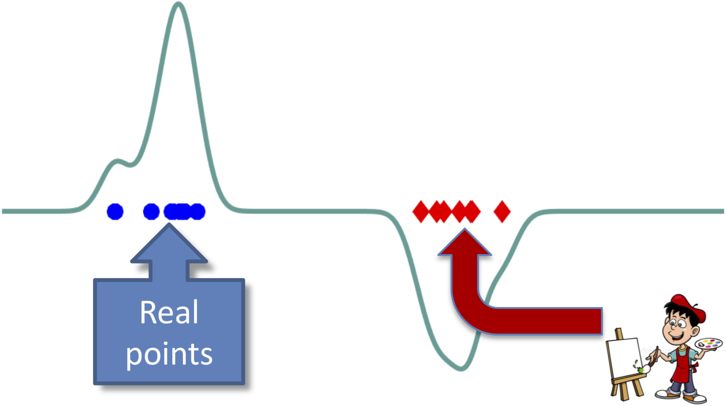
MMD=1.1



MMD as critic

An unhelpful critic witness:
$MMD(P, Q)$ with a narrow kernel.

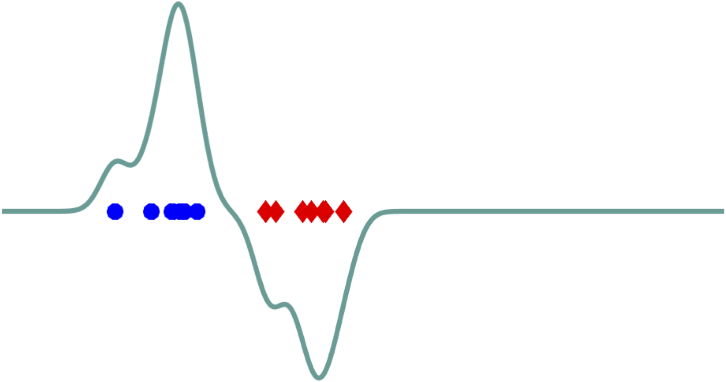
MMD=0.64



MMD as critic

An unhelpful critic witness:
$MMD(P, Q)$ with a narrow kernel.

MMD=0.64



MMD as GAN critic

From ICML 2015:

Generative Moment Matching Networks

Yujia Li[1] YUJIALI@CS.TORONTO.EDU
Kevin Swersky[1] KSWERSKY@CS.TORONTO.EDU
Richard Zemel[1,2] ZEMEL@CS.TORONTO.EDU
[1]Department of Computer Science, University of Toronto, Toronto, ON, CANADA
[2]Canadian Institute for Advanced Research, Toronto, ON, CANADA

From UAI 2015:

Training generative neural networks via Maximum Mean Discrepancy optimization

Gintare Karolina Dziugaite
University of Cambridge

Daniel M. Roy
University of Toronto

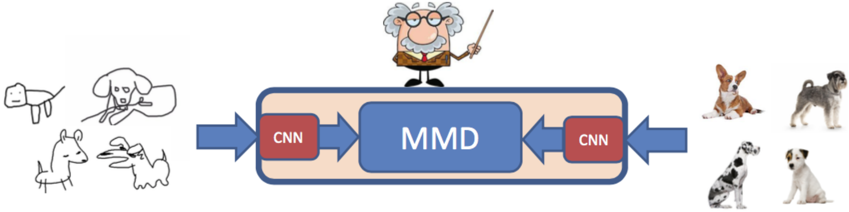
Zoubin Ghahramani
University of Cambridge

MMD as GAN critic

Need better image features.

CNN features for IPM witness functions

- Add convolutional features!
- The critic (teacher) also needs to be trained.



$\mathfrak{K}(x, y) = h_\psi{}^\top(x) h_\psi(y)$
where $h_\psi(x)$ is a CNN map:

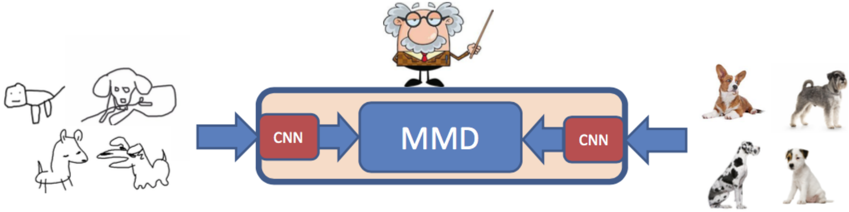
- Wasserstein GAN Arjovsky et al. [ICML 2017]
- WGAN-GP Gulrajani et al. [NeurIPS 2017]

$\mathfrak{K}(x, y) = k(h_\psi(x), h_\psi(y))$
where $h_\psi(x)$ is a CNN map,
k is e.g. an exponentiated quadratic kernel

MMD Li et al., [NeurIPS 2017]
Cramer Bellemare et al. [2017]
Coulomb Unterthiner et al., [ICLR 2018]
Demystifying MMD GANs Binkowski, Sutherland, Arbel, G., [ICLR 2018]

CNN features for IPM witness functions

- Add convolutional features!
- The critic (teacher) also needs to be trained.



$\mathfrak{K}(x, y) = {h_\psi}^\top(x) h_\psi(y)$
where $h_\psi(x)$ is a CNN map:

- Wasserstein GAN Arjovsky et al. [ICML 2017]
- WGAN-GP Gulrajani et al. [NeurIPS 2017]

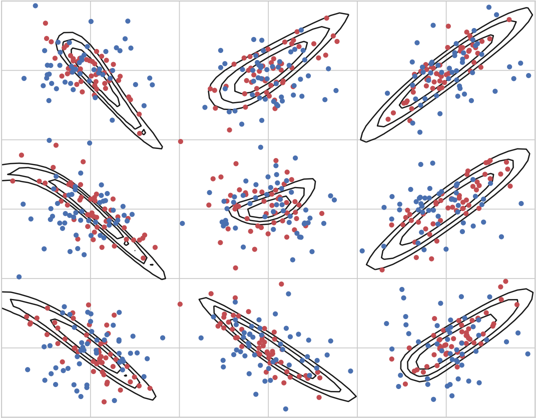
$\mathfrak{K}(x, y) = k(h_\psi(x), h_\psi(y))$
where $h_\psi(x)$ is a CNN map,
k is e.g. an exponentiated quadratic kernel
MMD Li et al., [NeurIPS 2017]
Cramer Bellemare et al. [2017]
Coulomb Unterthiner et al., [ICLR 2018]
Demystifying MMD GANs Binkowski, Sutherland, Arbel, G., [ICLR 2018]

Reminder: kernel with deep learned features

Kernel with deep learned features:

$$k_\theta(x, y) = [(1-\epsilon)\kappa(\Phi_\theta(x), \Phi_\theta(y)) + \epsilon]\, q(x, y)$$

κ and q are Gaussian kernels



Challenges for learned critic features

Learned critic features:

MMD with kernel $k(h_\psi(x), h_\psi(y))$ must give useful "gradient" to generator.

Challenges for learned critic features

Learned critic features:

MMD with kernel $k(h_\psi(x), h_\psi(y))$ must give useful "gradient" to generator.

Relation with test power?

If the MMD with kernel $k(h_\psi(x), h_\psi(y))$ gives a powerful test, will it be a good critic?

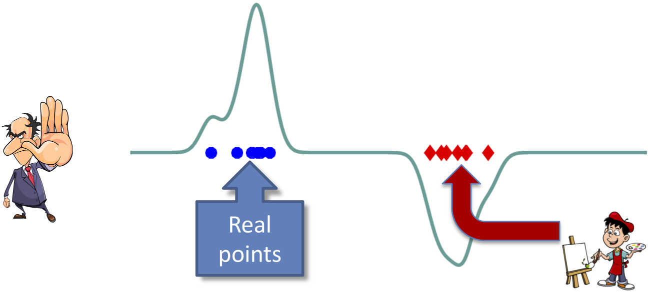
Challenges for learned critic features

Learned critic features:

MMD with kernel $k(h_\psi(x), h_\psi(y))$ must give useful "gradient" to generator.

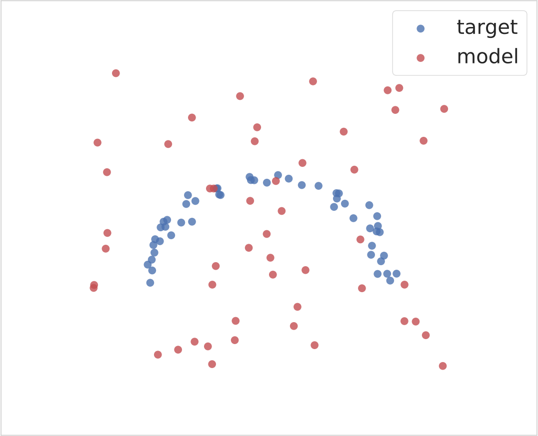
Relation with test power?

If the MMD with kernel $k(h_\psi(x), h_\psi(y))$ gives a powerful test, will it be a good critic?



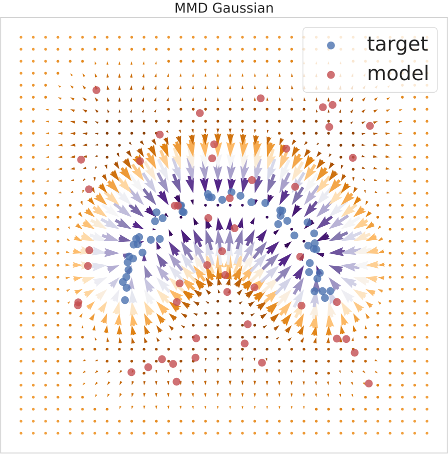
Simple 2-D example, *fixed* kernel

Samples from target P and model Q



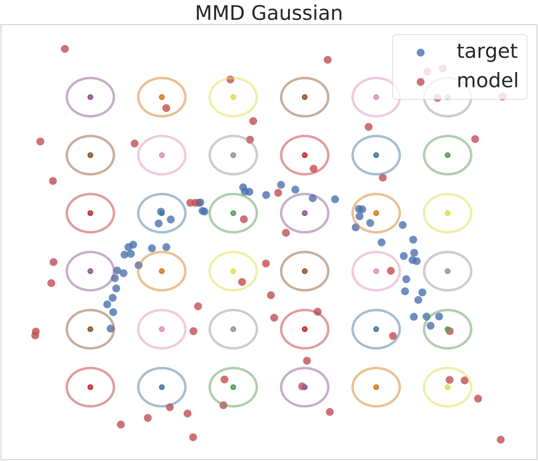
Simple 2-D example, *fixed* kernel

Witness gradient, MMD with exp. quad. kernel $k(x, y)$



Simple 2-D example, *fixed* kernel

What the kernels $k(x, y)$ look like



Adaptive neural net features + kernels

Use kernels $k(h_\psi(x), h_\psi(y))$ with features

$$h_\psi(x) = L_3\left(\left[\begin{array}{c} x \\ L_2(L_1(x)) \end{array}\right]\right)$$

where L_1, L_2, L_3 are fully connected with quadratic nonlinearity.

Adaptive neural net features + kernels

Witness gradient, maximize regularized $SMMD(P, \lambda)$
to learn $h_\psi(x)$ for $k(h_\psi(x), h_\psi(y))$

vector field movie, use Acrobat Reader to play

Adaptive neural net features + kernels

What the kenels $k(h_\psi(x), h_\psi(y))$ look like

isolines movie, use Acrobat Reader to play

A data-adaptive gradient penalty: NeurIPS 2018

- Gradient regulariser Arbel, Sutherland, Binkowski, G. [NeurIPS 2018]
- Also related to Sobolev GAN Mroueh et al. [ICLR 2018]

On gradient regularizers for MMD GANs

Michael Arbel
Gatsby Computational Neuroscience Unit
University College London
michael.n.arbel@gmail.com

Dougal J. Sutherland
Gatsby Computational Neuroscience Unit
University College London
dougal@gmail.com

Mikołaj Bińkowski
Department of Mathematics
Imperial College London
mikbinkowski@gmail.com

Arthur Gretton
Gatsby Computational Neuroscience Unit
University College London
arthur.gretton@gmail.com

A data-adaptive gradient penalty: NeurIPS 2018

- Gradient regulariser Arbel, Sutherland, Binkowski, G. [NeurIPS 2018]
- Also related to Sobolev GAN Mroueh et al. [ICLR 2018]

Maximise scaled MMD over critic features:

$$SMMD(P, \lambda) = \sigma_{P,\lambda}\ MMD$$

where

$$\sigma^2_{P,\lambda} = \lambda + \int k(h_\psi(x), h_\psi(x))\, dP(x) + \sum_{i=1}^{d} \int \partial_i \partial_{i+d} k(h_\psi(x), h_\psi(x))\ dP(x)$$

Our empirical observations

Data-dependent gradient regularizer of critic

Similar regularization strategies apply in:

- WGAN-GP Gulrajani et al. [NeurIPS 2017]
- "Witness function" in f-GANs (next talk!) Roth et al [NeurIPS 2017, eq. 19 and 20]

Our empirical observations

Data-dependent gradient regularizer of critic

Similar regularization strategies apply in:

- WGAN-GP Gulrajani et al. [NeurIPS 2017]
- "Witness function" in f-GANs (next talk!) Roth et al [NeurIPS 2017, eq. 19 and 20]

Alternate critic and generator training:

- Weaker critics can give better signals to poor (early stage) generators.
- Incomplete training of the critic is also a regularisation strategy

Don't *just* use gradient regularizers!

Spectral norm regularizer (effectively smooths critic class; ICLR 2018):

SPECTRAL NORMALIZATION
FOR GENERATIVE ADVERSARIAL NETWORKS

Takeru Miyato[1], Toshiki Kataoka[1], Masanori Koyama[2], Yuichi Yoshida[3]
{miyato, kataoka}@preferred.jp
koyama.masanori@gmail.com
yyoshida@nii.ac.jp
[1]Preferred Networks, Inc. [2]Ritsumeikan University [3]National Institute of Informatics

Entropic regularizer (avoid mode collapse):

arXiv.org > stat > arXiv:1910.04302

Statistics > Machine Learning

[Submitted on 9 Oct 2019]

Prescribed Generative Adversarial Networks

Adji B. Dieng, Francisco J. R. Ruiz, David M. Blei, Michalis K. Titsias

Evaluation and experiments

Benchmarks for comparison (all from ICLR 2018)

Spectral Normalization for Generative Adversarial Networks

Takeru Miyato[1], Toshiki Kataoka[1], Masanori Koyama[2], Yuichi Yoshida[3]
{miyato, kataoka}@preferred.jp
oyama masanori@gmail.com
yos i.ac.jp
works, Inc. [2]Ritsumeikan University [3]National Institute of Informatics

We combine with scaled MMD

Sobolev GAN

Youssef Mroueh[†], Chun-Liang Li[∘,⋆], Tom Sercu[†,⋆], Anant Raj[◊,⋆] & Yu Cheng[†]
† IBM Research AI
∘ Carnegie Mellon University
◊ Max Planck Institute for Intelligent Systems
⋆ denotes Equal Contribution
{mroueh,chengyu}@us.ibm.com, chunlial@cs.cmu.edu,
tom.sercu1@ibm.com,anant.raj@tuebingen.mpg.de

Demystifying MMD GANs

Mikołaj Bińkowski*
Department of Mathematics
Imperial College London
mikbinkowski@gmail.com

Dougal J. Sutherland*, Michael Arbel & Arthur Gretton
Gatsb Computational Neuroscience Unit
Uni College London
,michael.n.arbel,arthur.gretton}@gmail.com

Our ICLR 2018 paper

Boundary-Seeking Generative Adversarial Networks

R Devon Hjelm*
MILA, University of Montréal, IVADO
erroneus@gmail.com

Athul Paul Jacob*
MILA, MSR, University of Waterloo
apjacob@edu.uwaterloo.ca

Tong Che
MILA, University of Montréal
tong.che@umontreal.ca

Adam Trischler
MSR
adam.trischler@microsoft.com

Kyunghyun Cho
New York University,
CIFAR Azrieli Global Scholar
kyunghyun.cho@nyu.edu

Yoshua Bengio
MILA, University of Montréal, CIFAR, IVADO
yoshua.bengio@umontreal.ca

Results: unconditional imagenet 64×64

KID scores:

- BGAN: 47
- SN-GAN: 44
- SMMD GAN: 35

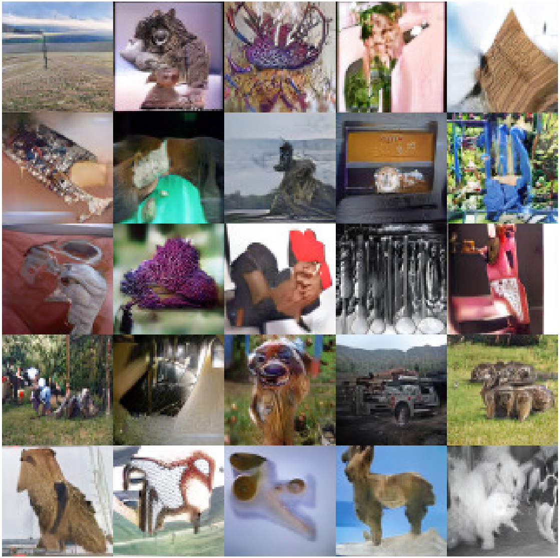
ILSVRC2012 (ImageNet) dataset, 1 281 167 images, resized to 64 × 64. 1000 classes.

Results: unconditional imagenet 64×64

KID scores:

- BGAN: 47
- SN-GAN: 44
- SMMD GAN: 35

ILSVRC2012 (ImageNet) dataset, 1 281 167 images, resized to 64 × 64. 1000 classes.



Results: unconditional imagenet 64×64

KID scores:

- BGAN: 47
- SN-GAN: 44
- SMMD GAN: 35

ILSVRC2012 (ImageNet) dataset, 1 281 167 images, resized to 64 × 64. 1000 classes.

Summary

- GAN critics rely on two sources of regularisation
  - Regularisation by incomplete training
  - Data-dependent gradient regulariser
- Some advantages of hybrid kernel/neural features:
  - MMD loss still a valid critic when features not optimal (unlike WGAN-GP)
  - Kernel features do some of the "work", so simpler h_ψ features possible.

"Demystifying MMD GANs," including KID score, ICLR 2018:

https://github.com/mbinkowski/MMD-GAN

Gradient regularised MMD, NeurIPS 2018:

https://github.com/MichaelArbel/Scaled-MMD-GAN

Linear vs nonlinear kenels

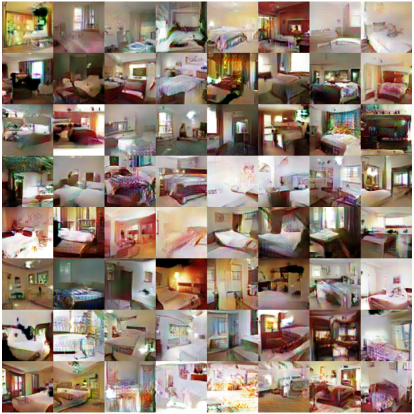
- Critic features from DCGAN: an f-filter critic has f, $2f$, $4f$ and $8f$ convolutional filters in layers 1-4. LSUN 64×64.

$k(h_\psi(x), h_\psi(y))$, $f = 64$, KID=3

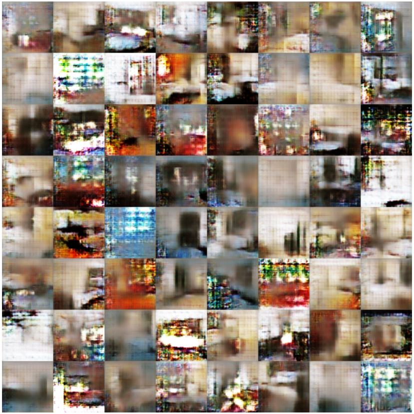
${h_\psi}^\top(x) h_\psi(y)$, $f = 64$, KID=4

Linear vs nonlinear kenels

- Critic features from DCGAN: an f-filter critic has f, $2f$, $4f$ and $8f$ convolutional filters in layers 1-4. LSUN 64×64.



$k(h_\psi(x), h_\psi(y))$, $f = 16$, KID=9



$h_\psi{}^\top(x) h_\psi(y)$, $f = 16$, KID=37

Evaluation of GANs

The inception score? Salimans et al. [NeurIPS 2016]

Based on the classification output $p(y|x)$ of the inception model Szegedy et al. [ICLR 2014],

$$E_X \exp KL(P(y|X)\|P(y)).$$

High when:

- predictive label distribution $P(y|x)$ has low entropy (good quality images)
- label entropy $P(y)$ is high (good variety).

Evaluation of GANs

The inception score? Salimans et al. [NeurIPS 2016]

Based on the classification output $p(y|x)$ of the inception model Szegedy et al. [ICLR 2014],

$$E_X \exp KL(P(y|X) \| P(y)).$$

High when:

- predictive label distribution $P(y|x)$ has low entropy (good quality images)
- label entropy $P(y)$ is high (good variety).

Problem: relies on a trained classifier! Can't be used on new categories (celeb, bedroom...)

Evaluation of GANs

The Frechet inception distance? Heusel et al. [NeurIPS 2017]

Fits Gaussians to features in the inception architecture (pool3 layer):

$$FID(P, Q) = \|\mu_P - \mu_Q\|^2 + \mathrm{tr}(\Sigma_P) + \mathrm{tr}(\Sigma_Q) - 2\mathrm{tr}\left((\Sigma_P \Sigma_Q)^{\frac{1}{2}}\right)$$

where μ_P and Σ_P are the feature mean and covariance of P

Evaluation of GANs

The Frechet inception distance? Heusel et al. [NeurIPS 2017]

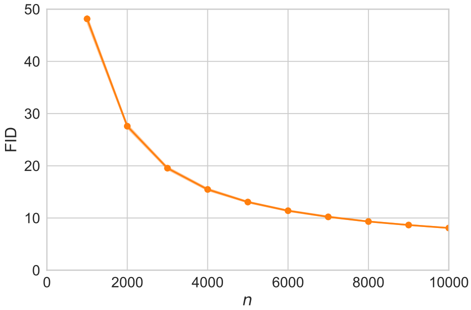
Fits Gaussians to features in the inception architecture (pool3 layer):

$$FID(P, Q) = \|\mu_P - \mu_Q\|^2 + \mathrm{tr}(\Sigma_P) + \mathrm{tr}(\Sigma_Q) - 2\mathrm{tr}\left((\Sigma_P \Sigma_Q)^{\frac{1}{2}}\right)$$

where μ_P and Σ_P are the feature mean and covariance of P

Problem: bias. For finite samples can consistently give incorrect answer.

- Bias demo, CIFAR-10 train vs test



Evaluation of GANs

The FID can give the wrong answer in theory.

Assume m samples from P and $n \to \infty$ samples from Q.

Given two alternatives:

$$P_1 \sim \mathcal{N}(0, (1 - m^{-1})^2) \qquad P_2 \sim \mathcal{N}(0, 1) \qquad Q \sim \mathcal{N}(0, 1).$$

Clearly,

$$FID(P_1, Q) = \frac{1}{m^2} > FID(P_2, Q) = 0$$

Given m samples from P_1 and P_2,

$$FID(\widehat{P_1}, Q) < FID(\widehat{P_2}, Q).$$

Evaluation of GANs

The FID can give the wrong answer in theory.

Assume m samples from P and $n \to \infty$ samples from Q.

Given two alternatives:

$$P_1 \sim \mathcal{N}(0, (1 - m^{-1})^2) \qquad P_2 \sim \mathcal{N}(0, 1) \qquad Q \sim \mathcal{N}(0, 1).$$

Clearly,

$$FID(P_1, Q) = \frac{1}{m^2} > FID(P_2, Q) = 0$$

Given m samples from P_1 and P_2,

$$FID(\widehat{P_1}, Q) < FID(\widehat{P_2}, Q).$$

Evaluation of GANs

The FID can give the wrong answer in theory.

Assume m samples from P and $n \to \infty$ samples from Q.

Given two alternatives:

$$P_1 \sim \mathcal{N}(0, (1 - m^{-1})^2) \qquad P_2 \sim \mathcal{N}(0, 1) \qquad Q \sim \mathcal{N}(0, 1).$$

Clearly,

$$FID(P_1, Q) = \frac{1}{m^2} > FID(P_2, Q) = 0$$

Given m samples from P_1 and P_2,

$$FID(\widehat{P_1}, Q) < FID(\widehat{P_2}, Q).$$

Evaluation of GANs

The FID can give the wrong answer in theory.

Assume m samples from P and $n \to \infty$ samples from Q.

Given two alternatives:

$$P_1 \sim \mathcal{N}(0, (1 - m^{-1})^2) \qquad P_2 \sim \mathcal{N}(0, 1) \qquad Q \sim \mathcal{N}(0, 1).$$

Clearly,

$$FID(P_1, Q) = \frac{1}{m^2} > FID(P_2, Q) = 0$$

Given m samples from P_1 and P_2,

$$FID(\widehat{P_1}, Q) < FID(\widehat{P_2}, Q).$$

Evaluation of GANs

The FID can give the wrong answer in practice.

Let $d = 2048$, and define

$$P_1 = \text{relu}(\mathcal{N}(0, I_d)) \quad P_2 = \text{relu}(\mathcal{N}(1, .8\Sigma + .2I_d)) \qquad Q = \text{relu}(\mathcal{N}(1, I_d))$$

where $\Sigma = \frac{4}{d}CC^T$, with C a $d \times d$ matrix with iid standard normal entries.

For a random draw of C:

$$FID(P_1, Q) \approx 1123.0 > 1114.8 \approx FID(P_2, Q)$$

With $m = 50\,000$ samples,

$$FID(\widehat{P_1}, Q) \approx 1133.7 < 1136.2 \approx FID(\widehat{P_2}, Q)$$

At $m = 100\,000$ samples, the ordering of the estimates is correct.

This behavior is similar for other random draws of C.

Evaluation of GANs

The FID can give the wrong answer in practice.

Let $d = 2048$, and define

$$P_1 = \text{relu}(\mathcal{N}(0, I_d)) \quad P_2 = \text{relu}(\mathcal{N}(1, .8\Sigma + .2I_d)) \quad Q = \text{relu}(\mathcal{N}(1, I_d))$$

where $\Sigma = \frac{4}{d} CC^T$, with C a $d \times d$ matrix with iid standard normal entries.

For a random draw of C:

$$FID(P_1, Q) \approx 1123.0 > 1114.8 \approx FID(P_2, Q)$$

With $m = 50\,000$ samples,

$$FID(\widehat{P_1}, Q) \approx 1133.7 < 1136.2 \approx FID(\widehat{P_2}, Q)$$

At $m = 100\,000$ samples, the ordering of the estimates is correct.

This behavior is similar for other random draws of C.

Evaluation of GANs

The FID can give the wrong answer in practice.

Let $d = 2048$, and define

$$P_1 = \mathrm{relu}(\mathcal{N}(0, I_d)) \quad P_2 = \mathrm{relu}(\mathcal{N}(1, .8\Sigma + .2I_d)) \quad Q = \mathrm{relu}(\mathcal{N}(1, I_d))$$

where $\Sigma = \frac{4}{d}CC^T$, with C a $d \times d$ matrix with iid standard normal entries.

For a random draw of C:

$$FID(P_1, Q) \approx 1123.0 > 1114.8 \approx FID(P_2, Q)$$

With $m = 50\,000$ samples,

$$FID(\widehat{P_1}, Q) \approx 1133.7 < 1136.2 \approx FID(\widehat{P_2}, Q)$$

At $m = 100\,000$ samples, the ordering of the estimates is correct.

This behavior is similar for other random draws of C.

Evaluation of GANs

The FID can give the wrong answer in practice.

Let $d = 2048$, and define

$$P_1 = \text{relu}(\mathcal{N}(0, I_d)) \quad P_2 = \text{relu}(\mathcal{N}(1, .8\Sigma + .2I_d)) \qquad Q = \text{relu}(\mathcal{N}(1, I_d))$$

where $\Sigma = \frac{4}{d}CC^T$, with C a $d \times d$ matrix with iid standard normal entries.

For a random draw of C:

$$FID(P_1, Q) \approx 1123.0 > 1114.8 \approx FID(P_2, Q)$$

With $m = 50\,000$ samples,

$$FID(\widehat{P_1}, Q) \approx 1133.7 < 1136.2 \approx FID(\widehat{P_2}, Q)$$

At $m = 100\,000$ samples, the ordering of the estimates is correct.

This behavior is similar for other random draws of C.

The kernel inception distance (KID)

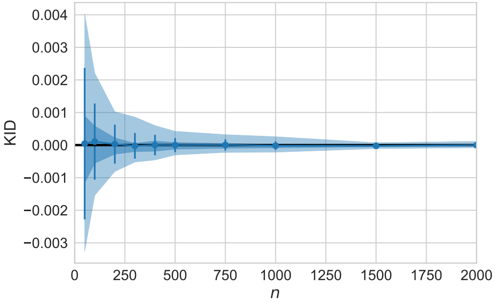
The Kernel inception distance Binkowski, Sutherland, Arbel, G. [ICLR 2018]

Measures similarity of the samples' representations in the inception architecture (pool3 layer)

MMD with kernel

$$k(x, y) = \left(\frac{1}{d} x^\top y + 1\right)^3 .$$

- Checks match for feature means, variances, skewness
- Unbiased : eg CIFAR-10 train/test



The kernel inception distance (KID)

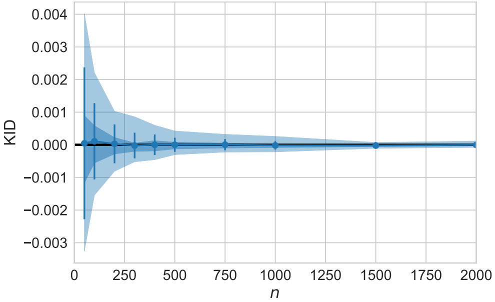
The Kernel inception distance Binkowski, Sutherland, Arbel, G. [ICLR 2018]

Measures similarity of the samples' representations in the inception architecture (pool3 layer)

MMD with kernel

$$k(x, y) = \left(\frac{1}{d} x^\top y + 1\right)^3.$$

- Checks match for feature means, variances, skewness
- Unbiased : eg CIFAR-10 train/test



..."but isn't KID is computationally costly?"

The kernel inception distance (KID)

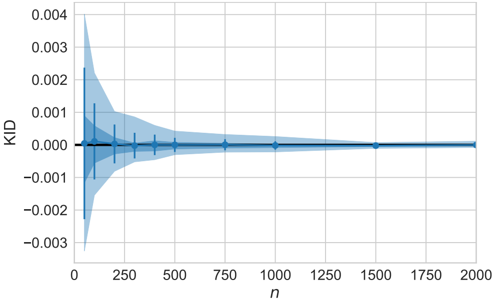
The Kernel inception distance Binkowski, Sutherland, Arbel, G. [ICLR 2018]

Measures similarity of the samples' representations in the inception architecture (pool3 layer)

MMD with kernel

$$k(x, y) = \left(\frac{1}{d} x^\top y + 1\right)^3.$$

- Checks match for feature means, variances, skewness
- Unbiased : eg CIFAR-10 train/test



..."but isn't KID is computationally costly?"

"Block" KID implementation is cheaper than FID: see paper (or use Tensorflow implementation)!

The kernel inception distance (KID)

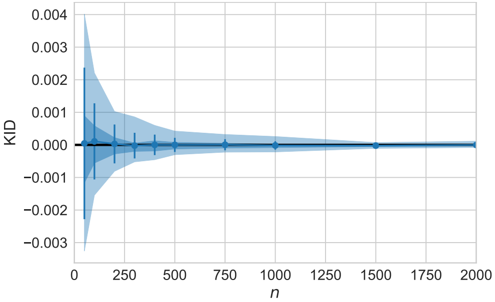
The Kernel inception distance Binkowski, Sutherland, Arbel, G. [ICLR 2018]

Measures similarity of the samples' representations in the inception architecture (pool3 layer)

MMD with kernel

$$k(x, y) = \left(\frac{1}{d}x^\top y + 1\right)^3.$$

- Checks match for feature means, variances, skewness
- Unbiased : eg CIFAR-10 train/test



Also used for automatic learning rate adjustment: if $KID(\widehat{P}_{t+1}, Q)$ not significantly better than $KID(\widehat{P}_t, Q)$ then reduce learning rate.

[Bounliphone et al. ICLR 2016]

Related: "An empirical study on evaluation metrics of generative adversarial networks", Xu et al. [arXiv, June 2018]