Two-Sample Testing
The problem

CIFAR-10 test set (Krizhevsky 2009)

\[ X \sim P \]

CIFAR-10.1 (Recht+ ICML 2019)

\[ Y \sim Q \]
The problem

CIFAR-10 test set (Krizhevsky 2009)  CIFAR-10.1 (Recht+ ICML 2019)

\[ X \sim P \]

\[ Y \sim Q \]

Are the distributions \( P \) and \( Q \) the same?
The problem

CIFAR-10 test set (Krizhevsky 2009)  CIFAR-10.1 (Recht+ ICML 2019)

\[ X \sim P \quad Y \sim Q \]

- Are the distributions \( P \) and \( Q \) the same?
- Remember that \( \text{MMD}(P, Q) = 0 \) iff \( P = Q \)
Estimating the MMD

\[
\text{MMD}(P, Q)^2 = \|\mu_P - \mu_Q\|^2
\]
Estimating the MMD

\[
\text{MMD}(P, Q)^2 = \|\mu_P - \mu_Q\|^2 \\
= \langle \mu_P, \mu_P \rangle - 2\langle \mu_P, \mu_Q \rangle + \langle \mu_Q, \mu_Q \rangle
\]
Estimating the MMD

\[ \text{MMD}(P, Q)^2 = \|\mu_P - \mu_Q\|^2 \]
\[ = \langle \mu_P, \mu_P \rangle - 2 \langle \mu_P, \mu_Q \rangle + \langle \mu_Q, \mu_Q \rangle \]
\[ = \mathbb{E} \left[ \langle \phi(X), \phi(X') \rangle - 2 \langle \phi(X), \phi(Y) \rangle + \langle \phi(Y), \phi(Y') \rangle \right] \]
Estimating the MMD

\[ MMD(P, Q)^2 = \|\mu_P - \mu_Q\|^2 \]

\[ = \langle \mu_P, \mu_P \rangle - 2\langle \mu_P, \mu_Q \rangle + \langle \mu_Q, \mu_Q \rangle \]

\[ = \mathbb{E} \left[\langle \varphi(X), \varphi(X') \rangle - 2\langle \varphi(X), \varphi(Y) \rangle + \langle \varphi(Y), \varphi(Y') \rangle\right] \]

\[ = \mathbb{E} \left[k(X, X') - 2k(X, Y) + k(Y, Y')\right] \]
Estimating the MMD

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

- Dogs (= \( P \)) and fish (= \( Q \)) example
- Each entry is one of \( k(\text{dog}_i, \text{dog}_j) \), \( k(\text{dog}_i, \text{fish}_j) \), or \( k(\text{fish}_i, \text{fish}_j) \)
- \( MMD(P, Q)^2 = \mathbb{E} \left[ k(\text{dog}_i, \text{dog}_j) + k(\text{fish}_i, \text{fish}_j) - 2k(\text{dog}_i, \text{fish}_j) \right] \)
Estimating the MMD

- Dogs ($= P$) and fish ($= Q$) example
- Each entry is one of $k(\text{dog}_i, \text{dog}_j)$, $k(\text{dog}_i, \text{fish}_j)$, or $k(\text{fish}_i, \text{fish}_j)$
- $MMD(P, Q)^2 = \mathbb{E} \left[ k(\text{dog}_i, \text{dog}_j) + k(\text{fish}_i, \text{fish}_j) - 2k(\text{dog}_i, \text{fish}_j) \right]$
Using a divergence estimator

CIFAR-10 test set (Krizhevsky 2009)

\[ X \sim P \]

Say we get \( \hat{MMD}^2 = 0.09116 \)

CIFAR-10.1 (Recht+ ICML 2019)

\[ Y \sim Q \]
Using a divergence estimator

CIFAR-10 test set (Krizhevsky 2009)

\[ X \sim P \]

\[ Y \sim Q \]

- Say we get \( \widehat{MMD}^2 = 0.09116 \)

CIFAR-10.1 (Recht+ ICML 2019)
Using a divergence estimator

CIFAR-10 test set (Krizhevsky 2009)

\[ X \sim P \]

CIFAR-10.1 (Recht+ ICML 2019)

\[ Y \sim Q \]

- Say we get \[ \hat{MMD}^2 = 0.09116 \]
- ...great. Is the true MMD zero? Equivalently: is \( P = Q \)?
Using a divergence estimator

CIFAR-10 test set (Krizhevsky 2009)

\[ X \sim P \]

CIFAR-10.1 (Recht+ ICML 2019)

\[ Y \sim Q \]

- Say we get \( \widehat{MMD}^2 = 0.09116 \)
- ...great. Is the true MMD zero? Equivalently: is \( P = Q \)?
- We need to know “how random” \( \widehat{MMD}^2 \) is...
Behavior of $\hat{MMD}^2$ when $P \neq Q$

- $P$, $Q$ Laplace with different variances in $y$
- Draw $n = 200$ i.i.d samples from $P$ and $Q$
Behavior of $\hat{\text{MMD}}^2$ when $P \neq Q$

- $P$, $Q$ Laplace with different variances in $y$
- Draw $n = 200$ i.i.d samples from $P$ and $Q$

Number of MMDs: 1

\[
\sqrt{n} \times \hat{\text{MMD}}^2 = 1.2
\]
Behavior of $\widehat{MMD}^2$ when $P \neq Q$

- $P$, $Q$ Laplace with different variances in $y$
- Draw $n = 200$ new i.i.d. samples from $P$ and $Q$

Number of MMDs: 2

$\sqrt{n} \times \widehat{MMD}^2 = 1.5$
Behavior of $\text{MMD}^2$ when $P \neq Q$

- $P$, $Q$ Laplace with different variances in $y$
- Draw $n = 200$ i.i.d samples from $P$ and $Q$, 150 times

Number of MMDs: 150

![Histogram of $\sqrt{n} \times \text{MMD}^2$](image)
Behavior of $\sqrt{n} \times \hat{MMD}^2$ when $P \neq Q$

- $P$, $Q$ Laplace with different variances in $y$
- Draw $n = 200$ i.i.d samples from $P$ and $Q$, 300 times

Number of MMDs: 300
Behavior of $\hat{MMD}^2$ when $P \neq Q$

- $P, Q$ Laplace with different variances in $y$
- Draw $n = 200$ i.i.d samples from $P$ and $Q$, 3000 times

Number of MMDs: 3000
Asymptotics of $\widehat{\text{MMD}}^2$ when $P \neq Q$

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

$$\sqrt{n} \frac{\widehat{\text{MMD}}^2 - \text{MMD}(P, Q)}{\sigma_{H_1}} \xrightarrow{D} \mathcal{N}(0, 1),$$

where $\sigma_{H_1}^2/n$ is asymptotic variance (depends on $P$, $Q$, $k$).
Behavior of $\overrightarrow{MMD}^2$ when $P = Q$

What about when $P$ and $Q$ are the same?
Behavior of $\hat{MMD}^2$ when $P = Q$

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

Number of MMDs: 10
Behavior of $\widehat{MMD}^2$ when $P = Q$

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

Number of MMDs: 20
Behavior of $\overrightarrow{MMD}^2$ when $P = Q$

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

Number of MMDs: 50
Behavior of $\hat{MMD}^2$ when $P = Q$

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

Number of MMDs: 100

![Graph showing the distribution of $n \times \hat{MMD}^2$]
Behavior of $\widehat{MMD}^2$ when $P = Q$

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

Number of MMDs: 1000
Asymptotics of $\hat{MMD}^2$ when $P = Q$

Where $P = Q$, statistic has asymptotic distribution

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

where

$$\lambda_l \psi_i(x') = \int_{\mathcal{X}} \tilde{k}(x, x') \psi_i(x) dP(x)$$

centered

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

A summary of the asymptotics:
Test construction: (Gretton+., JMLR 2012)

![Graph illustrating statistical testing](image-url)
Statistical testing

Test construction: (Gretton+, JMLR 2012)

- don't reject $H_0$
- reject $H_0$ (say $P \neq Q$)

Probability density

- $P = Q$
- $P \neq Q$

False rejection rate: want $\leq \alpha$
Statistical testing

Test construction: (Gretton+ , JMLR 2012)

- don't reject $H_0$
- reject $H_0$ (say $P \neq Q$)

- false rejection rate: want $\leq \alpha$
- power: true rejection rate

$n \overset{\text{MMD}^2}{\rightarrow} (X, Y)$

- probability density
- $C_\alpha$
How do we get the test threshold $c_\alpha$?

Original empirical MMD for dogs and fish:

$$\mathbf{X} = \begin{bmatrix} \text{dog} & \text{dog} & \text{dog} & \ldots \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} \text{fish} & \text{fish} & \text{fish} & \ldots \end{bmatrix}$$

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

$$+ \frac{1}{n(n-1)} \sum_{i \neq j} k(y_i, y_j)$$

$$- \frac{2}{n^2} \sum_{i,j} k(x_i, y_j)$$
How do we get the test threshold $c_\alpha$?

Permuted **dog** and **fish** samples (**merdogs**):

\[
\tilde{X} = [\text{fish} \quad \text{dog} \quad \text{...}] \\
\tilde{Y} = [\text{dog} \quad \text{fish} \quad \text{...}]
\]
How do we get the test threshold $c_\alpha$?

Permuted dog and fish samples (merdogs):

$\tilde{X} = [\text{dog, fish, ...}]$

$\tilde{Y} = [\text{dog, fish, ...}]$

$\overline{\text{MMD}}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(\tilde{x}_i, \tilde{x}_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\tilde{y}_i, \tilde{y}_j) - \frac{2}{n^2} \sum_{i,j} k(\tilde{x}_i, \tilde{y}_j)$
How do we get the test threshold $c_\alpha$?

Permuted dog and fish samples (merdogs):

$$\tilde{X} = \begin{bmatrix}
\text{fish} & \text{dog} & \text{fish} & \ldots \\
\end{bmatrix}$$

$$\tilde{Y} = \begin{bmatrix}
\text{dog} & \text{fish} & \text{dog} & \ldots \\
\end{bmatrix}$$

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

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

$$- \frac{2}{n^2} \sum_{i,j} k(\tilde{x}_i, \tilde{y}_j)$$
How do we get the test threshold $c_\alpha$?

Permuted dog and fish samples (merdogs):

\[
\hat{X} = \begin{bmatrix}
\text{fish} & \text{dog} & \text{fish} & \ldots \\
\end{bmatrix}
\]

\[
\hat{Y} = \begin{bmatrix}
\text{dog} & \text{fish} & \text{dog} & \ldots \\
\end{bmatrix}
\]

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

- This simulates $P = Q$
How do we get the test threshold $c_\alpha$?

Permuted dog and fish samples (merdogs):

\[
\tilde{X} = \begin{bmatrix}
\text{fish} & \text{dog} & \text{fish} & \ldots
\end{bmatrix}
\]

\[
\tilde{Y} = \begin{bmatrix}
\text{dog} & \text{fish} & \text{dog} & \ldots
\end{bmatrix}
\]

\[
\overline{\text{MMD}}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(\tilde{x}_i, \tilde{x}_j)
\]

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

\[
- \frac{2}{n^2} \sum_{i,j} k(\tilde{x}_i, \tilde{y}_j)
\]

- This simulates $P = Q$
- Repeat, set $c_\alpha$ to quantile
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 \( \sigma \): for any \( P \) and \( Q \), power \( \to 1 \) as \( n \to \infty \)
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 \( \sigma \): for any \( P \) and \( Q \), power \( \to 1 \) as \( n \to \infty \)
- But choice of \( \sigma \) 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 \( \sigma \): for any \( P \) and \( Q \), power \( \to 1 \) as \( n \to \infty \)
- But choice of \( \sigma \) is very important for finite \( n \)...

![Graph showing the kernel function](image)
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 \( \sigma \): for any \( P \) and \( Q \), power \( \rightarrow 1 \) as \( n \rightarrow \infty \)
- But choice of \( \sigma \) 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 \( \sigma \): for any \( P \) and \( Q \), power \( \to 1 \) as \( n \to \infty \)
- But choice of \( \sigma \) 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 \( \sigma \): for any \( P \) and \( Q \), power \( \to 1 \) as \( n \to \infty \)
- But choice of \( \sigma \) is very important for finite \( n \)...
- ...and some problems (e.g. images) might have no good choice for \( \sigma \)
Choosing a kernel for the test

- Often helpful to use a relevant representation $\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$, eg:

$$k(x, y) = k_{\text{top}}(\Phi(x), \Phi(y))$$
Choosing a kernel for the test

- Often helpful to use a relevant representation $\Phi : \mathcal{X} \to \mathbb{R}^d$, eg:

$$k(x, y) = k_{\text{top}}(\Phi(x), \Phi(y))$$

- Take $\Phi$ as predictions of a pretrained classifier on a related domain
Choosing a kernel for the test

- Often helpful to use a relevant representation $\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$, eg:

  $$k(x, y) = k_{\text{top}}(\Phi(x), \Phi(y))$$

- Take $\Phi$ as predictions of a pretrained classifier on a related domain
  - Related to Adversarial Accuracy (Yang+ ICLR 2017) and Inception Score (Salimans+ NeurIPS 2016).
Choosing a kernel for the test

- Often helpful to use a relevant representation \( \Phi : \mathcal{X} \rightarrow \mathbb{R}^d \), eg:

\[
k(x, y) = k_{\text{top}}(\Phi(x), \Phi(y))
\]

- Take \( \Phi \) as predictions of a pretrained classifier on a related domain
  - Related to Adversarial Accuracy (Yang+ ICLR 2017) and Inception Score (Salimans+ NeurIPS 2016). \textit{We’ll come back to this!}
Choosing a kernel for the test

- Often helpful to use a relevant representation $\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$, eg:

$$k(x, y) = k_{\text{top}}(\Phi(x), \Phi(y))$$

- Take $\Phi$ as predictions of a pretrained classifier on a related domain
  - Related to Adversarial Accuracy (Yang+ ICLR 2017) and Inception Score (Salimans+ NeurIPS 2016). *We’ll come back to this!*
- Take $\Phi$ as late hidden layer from pretrained related classifier
  - KID (Bińkowski, Sutherland+ ICLR 2018), Xu+ (arXiv:1806.07755)
Choosing a kernel for the test

- Often helpful to use a relevant representation $\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$, eg:

$$k(x, y) = k_{\text{top}}(\Phi(x), \Phi(y))$$

- Take $\Phi$ as predictions of a pretrained classifier on a related domain
  - Related to Adversarial Accuracy (Yang+ ICLR 2017) and Inception Score (Salimans+ NeurIPS 2016). We’ll come back to this!
- Take $\Phi$ as late hidden layer from pretrained related classifier
  - KID (Bińkowski, Sutherland+ ICLR 2018), Xu+ (arXiv:1806.07755)
  - Closely related to FID (Heusel+ NeurIPS 2017) but much nicer statistical properties, more correlated with human judgement (Zhou, Gordon+ NeurIPS 2019)
Choosing a kernel for the test

- Bau et al. (ICCV 2019) compare counts of pixel categories

![Graph](image.png)

(a) generated vs training object segmentation statistics
What about tests for other distances?

- Sometimes, nice closed forms for threshold (like a \( t \) test)
- Asymptotic behavior of KALE, Wasserstein, \ldots mostly unknown
- But permutation tests usually work!
Choosing the best test
A test’s power depends on $P$ and $Q$ (and $n$)

Many MMDs have 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 $P$ and $Q$ (and $n$)
- Many MMDs have power $\to 1$ as $n \to \infty$ for any (fixed) problem
  - But, for many $P$ and $Q$, will have terrible power with reasonable $n$!
- Can maybe pick a good kernel manually for a given problem
- Can’t get one that has good finite-sample power for all problems
  - No one test can have all that power
Choosing test power

- Best test (of level $\alpha$) is the one with highest test power

\[ n \tilde{\text{MMD}}^2(X, Y) \]

Probability density

- don't reject $H_0$
- reject $H_0$ (say $P \neq Q$)

false rejection rate: want \( \leq \alpha \)

power: true rejection rate

\[ C_\alpha \]
Optimizing MMD for test power

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

$$Pr_1(nMMD^2 > \hat{c}_\alpha)$$

- $\hat{c}_\alpha$ is an estimate of the test threshold $c_\alpha$
Optimizing MMD for test power

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

$$\Pr_1\left(n \hat{MMD}^2 > \hat{c}_\alpha\right) = \Pr_1\left(\sqrt{n} \frac{\hat{MMD}^2 - MMD^2}{\sigma_{H_1}} > \frac{\hat{c}_\alpha}{\sqrt{n\sigma_{H_1}}} - \sqrt{nMMD^2}\frac{\sigma_{H_1}}{\sigma_{H_1}}\right)$$

- $\hat{c}_\alpha$ is an estimate of the test threshold $c_\alpha$
Optimizing MMD for test power

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

$$Pr_1\left(n\hat{MMD}^2 > \hat{c}_\alpha\right) = Pr_1\left(\sqrt{n}\frac{\hat{MMD}^2 - MMD^2}{\sigma_{H_1}} > \frac{\hat{c}_\alpha}{\sqrt{n}\sigma_{H_1}} - \frac{\sqrt{n}MMD^2}{\sigma_{H_1}}\right)$$

$$\rightarrow \Phi\left(\sqrt{n}\frac{MMD^2}{\sigma_{H_1}} - \frac{c_\alpha}{\sqrt{n}\sigma_{H_1}}\right)$$

- $\hat{c}_\alpha$ is an estimate of the test threshold $c_\alpha$
- $\Phi$ is the CDF of the standard normal distribution
Optimizing MMD for test power

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

$$
\Pr_1 \left( n \hat{\text{MMD}}^2 > \hat{c}_\alpha \right) = \Pr_1 \left( \sqrt{n} \frac{\hat{\text{MMD}}^2 - \text{MMD}^2}{\sigma_{H_1}} > \frac{\hat{c}_\alpha}{\sqrt{n} \sigma_{H_1}} - \frac{\sqrt{n} \text{MMD}^2}{\sigma_{H_1}} \right)
$$

\[ \to \Phi \left( \sqrt{n} \frac{\text{MMD}^2}{\sigma_{H_1}} - \frac{c_\alpha}{\sqrt{n} \sigma_{H_1}} \right) \]

- For large $n$, second term is negligible!
Optimizing MMD for test power

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

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

$$= \Pr_1 \left( \sqrt{n} \frac{\hat{MMD}^2 - MMD^2}{\sigma_{H_1}} > \frac{\hat{c}_\alpha}{\sqrt{n} \sigma_{H_1}} - \frac{\sqrt{n} MMD^2}{\sigma_{H_1}} \right)$$

$$\to \Phi \left( \sqrt{n} \frac{MMD^2}{\sigma_{H_1}} - \frac{c_\alpha}{\sqrt{n} \sigma_{H_1}} \right)$$

■ To maximize test power, choose $k$ to maximize (Sutherland+ ICLR 2017)

$$\frac{MMD^2(P, Q)}{\sigma_{H_1}(P, Q)}$$

• Estimator is differentiable in kernel parameters!
Data splitting

Choose a kernel $k$ maximizing $\frac{\text{MMD}^2}{\hat{\sigma}_{H_1}}$

Use chosen $k$ for MMD test
Learning a kernel helps a lot

- Even just learning a bandwidth... (Sutherland+ ICLR 2017)
Learning a kernel helps a lot

- Even just learning a bandwidth... (Sutherland+ ICLR 2017)
- ...but you can learn a lot more: $k_\theta(x, y) = k_{\text{top}}(\Phi_\theta(x), \Phi_\theta(y))$
Learning a kernel helps a lot

- Even just learning a bandwidth... (Sutherland+ ICLR 2017)
- ...but you can learn a lot more: $k_\theta(x, y) = k_{\text{top}}(\Phi_\theta(x), \Phi_\theta(y))$
  - Learning a deep kernel for CIFAR-10 vs CIFAR-10.1 rejects the null

CIFAR-10 test set (Krizhevsky 2009)

\[ X \sim P \]

CIFAR-10.1 (Recht+ ICML 2019)

\[ Y \sim Q \]
Alternative approach: Classifier two-sample tests

- Train a classifier $f : \mathcal{X} \rightarrow \{1, -1\}$ on $P$ from $Q$
- Test statistic: accuracy on test set (Lopez-Paz and Oquab, ICLR 2017)

Train a classifier $f$

Evaluate accuracy of $f$

$X \sim P$

$Y \sim Q$
Alternative approach: Classifier two-sample tests

- Train a classifier \( f : \mathcal{X} \rightarrow \{1, -1\} \) on \( P \) from \( Q \)
- Test statistic: accuracy on test set (Lopez-Paz and Oquab, ICLR 2017)
- Almost exactly equivalent:

\[
    k_f(x, y) = \frac{1}{4} \mathbb{1}(f(x) > 0) \mathbb{1}(f(y) > 0)
\]

gives

\[
    MMD(P, Q) = \left| \text{accuracy} - \frac{1}{2} \right|
\]
Alternative approach: Classifier two-sample tests

- Train a classifier \( f : \mathcal{X} \to \{1, -1\} \) on \( P \) from \( Q \)
- Test statistic: accuracy on test set (Lopez-Paz and Oquab, ICLR 2017)
- Almost exactly equivalent:

\[
  k_f(x, y) = \frac{1}{4} \mathbb{1}(f(x) > 0) \mathbb{1}(f(y) > 0)
\]

gives

\[
  \text{MMD}(P, Q) = \left| \text{accuracy} - \frac{1}{2} \right|
\]

- \( \sigma_{H_1} \) decreases with acc: maximizing \( \frac{\text{MMD}^2}{\sigma_{H_1}} \) exactly maximizes power
Alternative approach: Classifier two-sample tests

- Train a classifier $f : \mathcal{X} \rightarrow \{1, -1\}$ on $P$ from $Q$
- Test statistic: accuracy on test set (Lopez-Paz and Oquab, ICLR 2017)
- Almost exactly equivalent:

$$k_f(x, y) = \frac{1}{4} \mathbb{1}(f(x) > 0) \mathbb{1}(f(y) > 0)$$

which gives

$$MMD(P, Q) = \left| \text{accuracy} - \frac{1}{2} \right|$$

- 0-1 kernel inflates variance, decreases test power
Alternative approach: Classifier two-sample tests

- Train a classifier $f : \mathcal{X} \rightarrow \{1, -1\}$ on $P$ from $Q$
- Test statistic: accuracy on test set (Lopez-Paz and Oquab, ICLR 2017)
- Almost exactly equivalent:

$$k_f(x, y) = \frac{1}{4} \mathbb{1}(f(x) > 0) \mathbb{1}(f(y) > 0)$$

gives

$$MMD(P, Q) = \left| \text{accuracy} - \frac{1}{2} \right|$$

- 0-1 kernel inflates variance, decreases test power
  - Intermediate option: $k(x, y) = f(x) f(y)$
Alternative approach: Classifier two-sample tests

- Train a classifier \( f : \mathcal{X} \to \{1, -1\} \) on \( P \) from \( Q \)
- Test statistic: accuracy on test set (Lopez-Paz and Oquab, ICLR 2017)
- Almost exactly equivalent:

\[
k_f(x, y) = \frac{1}{4} \mathbb{1}(f(x) > 0) \mathbb{1}(f(y) > 0)
\]

gives

\[
MMD(P, Q) = \left| \text{accuracy} - \frac{1}{2} \right|
\]

- 0-1 kernel inflates variance, decreases test power
  - Intermediate option: \( k(x, y) = f(x) f(y) \)

- Empricially: deep kernel > linear > 0-1
Alternative approach: Classifier two-sample tests

- Train a classifier $f : \mathcal{X} \rightarrow \{1, -1\}$ on $P$ from $Q$
- Test statistic: accuracy on test set (Lopez-Paz and Oquab, ICLR 2017)
- Almost exactly equivalent:

$$k_f(x, y) = \frac{1}{4} \mathbb{1}(f(x) > 0) \mathbb{1}(f(y) > 0)$$

gives

$$MMD(P, Q) = \left| \text{accuracy} - \frac{1}{2} \right|$$

- 0-1 kernel inflates variance, decreases test power
  - Intermediate option: $k(x, y) = f(x) f(y)$
- Also trains for cross-entropy, instead of power directly(ish)
- Empirically: deep kernel $> \text{linear} > 0$-1, $\frac{\widehat{MMD}^2}{\hat{\sigma}_{H_1}} > \text{cross-entropy}$
Interpreting the learned kernel

MNIST samples

Samples from a GAN
Interpreting the learned kernel

\[ k(\mathbf{4}, \mathbf{2}) = \prod_{i=1}^{D} \exp \left( -\frac{(\mathbf{4}[i] - \mathbf{2}[i])^2}{\sigma_i^2} \right) \]
Interpreting the learned kernel

MNIST samples

Samples from a GAN

- Power for optimized ARD kernel: 1.00 at $\alpha = 0.01$
- Power for optimized RBF kernel: 0.57 at $\alpha = 0.01$
Interpreting points with largest witness function values

(Sutherland+ ICLR 2017)
Interpreting points with largest witness function values

Prototypes

Criticisms

(Kim+ NeurIPS 2016)
Main references and further reading

- **MMD asymptotics and test construction:**

- **Kernels for tests on images:**
  - Bińkowski, Sutherland, Arbel, Gretton. Demystifying MMD GANs (2018)

- **Another approach: random 1d projection is almost surely consistent**
  - Heller, Heller. Multivariate tests of association based on univariate tests (2016)

- **Optimizing test kernels / classifiers:**
  - Sutherland, Tung, Strathmann, De, Ramdas, Smola, Gretton. Generative Models and Model Criticism via Optimized Maximum Mean Discrepancy (2017)
    - Also our not-quite-on-arXiv-yet followup... (with Feng Liu, Wenkai Xu, Jie Lu, Guangquang Zhang)

- **Interpreting via witness functions:**
  - Kim, Khanna, Koyejo. Examples are not Enough, Learn to Criticize! Criticism for Interpretability (2016)