An Adaptive Test of Independence with Analytic Kernel Embeddings

Wittawat Jitkrittum
Gatsby Unit, University College London
wittawat@gatsby.ucl.ac.uk

Probabilistic Graphical Model Workshop 2017
Institute of Statistical Mathematics, Tokyo

24 Feb 2017
Reference

Coauthors:

Arthur Gretton
Gatsby Unit, UCL

Zoltán Szabó
École Polytechnique

Preprint:

An Adaptive Test of Independence with Analytic Kernel Embeddings
Wittawat Jitkrittum, Zoltán Szabó, Arthur Gretton
https://arxiv.org/abs/1610.04782

Python code: https://github.com/wittawatj/fsic-test
What Is Independence Testing?

- Let \( X \in \mathbb{R}^{d_x}, Y \in \mathbb{R}^{d_y} \) be random vectors following \( P_{xy} \).
- Given a joint sample \( \{(x_i, y_i)\}_{i=1}^{n} \sim P_{xy} \) (unknown), test
  \[
  H_0 : P_{xy} = P_x P_y, \\
  \text{vs. } H_1 : P_{xy} \neq P_x P_y.
  \]

- \( P_{xy} = P_x P_y \) equivalent to \( X \perp Y \).
- Compute a test statistic \( \hat{\lambda}_n \). Reject \( H_0 \) if \( \hat{\lambda}_n \geq T_\alpha \) (threshold).
- \( T_\alpha = (1 - \alpha) \)-quantile of the null distribution.
What Is Independence Testing?

- Let $X \in \mathbb{R}^{d_x}, Y \in \mathbb{R}^{d_y}$ be random vectors following $P_{xy}$.
- Given a joint sample $\{(x_i, y_i)\}_{i=1}^n \sim P_{xy}$ (unknown), test
  
  $H_0 : P_{xy} = P_x P_y,$
  
  vs. $H_1 : P_{xy} \neq P_x P_y.$

- $P_{xy} = P_x P_y$ equivalent to $X \perp Y$.
- Compute a test statistic $\hat{\lambda}_n$. Reject $H_0$ if $\hat{\lambda}_n \geq T_\alpha$ (threshold).
- $T_\alpha = (1 - \alpha)$-quantile of the null distribution.
What Is Independence Testing?

- Let \( X \in \mathbb{R}^{d_x}, Y \in \mathbb{R}^{d_y} \) be random vectors following \( P_{xy} \).
- Given a joint sample \( \{(x_i, y_i)\}_{i=1}^{n} \sim P_{xy} \) (unknown), test

\[
H_0 : P_{xy} = P_x P_y, \\
vs. \ H_1 : P_{xy} \neq P_x P_y.
\]

- \( P_{xy} = P_x P_y \) equivalent to \( X \perp Y \).
- Compute a test statistic \( \hat{\lambda}_n \). Reject \( H_0 \) if \( \hat{\lambda}_n \geq T_\alpha \) (threshold).
- \( T_\alpha = (1 - \alpha) \)-quantile of the null distribution.

\[
\begin{align*}
P^\mathcal{H}_0(\hat{\lambda}_n) & \\
P^\mathcal{H}_1(\hat{\lambda}_n) & \\
T_\alpha & \\
\hat{\lambda}_n &
\end{align*}
\]
What Is Independence Testing?

- Let $X \in \mathbb{R}^{d_x}$, $Y \in \mathbb{R}^{d_y}$ be random vectors following $P_{xy}$.
- Given a joint sample $\{(x_i, y_i)\}_{i=1}^n \sim P_{xy}$ (unknown), test

  $H_0 : P_{xy} = P_x P_y,$

  vs. $H_1 : P_{xy} \neq P_x P_y.$

- $P_{xy} = P_x P_y$ equivalent to $X \perp Y$.
- Compute a test statistic $\hat{\lambda}_n$. Reject $H_0$ if $\hat{\lambda}_n \geq T_\alpha$ (threshold).
- $T_\alpha = (1 - \alpha)$-quantile of the null distribution.
Want a test which is ...

1. **Non-parametric** i.e., no parametric assumption on $P_{xy}$.
2. **Linear-time** i.e., computational complexity is $O(n)$. Fast.
3. **Adaptive** i.e., has a well-defined criterion for parameter tuning.

<table>
<thead>
<tr>
<th></th>
<th>Non-parametric</th>
<th>$O(n)$</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson correlation</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HSIC [Gretton et al., 2005]</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HSIC with RFFs* [Zhang et al., 2016]</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>FSIC (proposed)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

*: RFFs = Random Fourier Features

- Focus on cases where $n$ (sample size) is large.
Goals

Want a test which is …

1. **Non-parametric** i.e., no parametric assumption on $P_{xy}$.
2. **Linear-time** i.e., computational complexity is $\mathcal{O}(n)$. Fast.
3. **Adaptive** i.e., has a well-defined criterion for parameter tuning.

<table>
<thead>
<tr>
<th></th>
<th>Non-parametric</th>
<th>$\mathcal{O}(n)$</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson correlation</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HSIC [Gretton et al., 2005]</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HSIC with RFFs* [Zhang et al., 2016]</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>FSIC (proposed)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* RFFs = Random Fourier Features

Focus on cases where $n$ (sample size) is large.
Goals

Want a test which is …

1. **Non-parametric** i.e., no parametric assumption on $P_{xy}$.
2. **Linear-time** i.e., computational complexity is $O(n)$. Fast.
3. **Adaptive** i.e., has a well-defined criterion for parameter tuning.

<table>
<thead>
<tr>
<th></th>
<th>Non-parametric</th>
<th>$O(n)$</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson correlation</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HSIC [Gretton et al., 2005]</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>HSIC with RFFs* [Zhang et al., 2016]</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>FSIC (proposed)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* : RFFs = Random Fourier Features

- Focus on cases where $n$ (sample size) is large.
Goals

Want a test which is …

1. **Non-parametric** i.e., no parametric assumption on $P_{xy}$.
2. **Linear-time** i.e., computational complexity is $\mathcal{O}(n)$. Fast.
3. **Adaptive** i.e., has a well-defined criterion for parameter tuning.

<table>
<thead>
<tr>
<th>Test</th>
<th>Non-parametric</th>
<th>$\mathcal{O}(n)$</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson correlation</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HSIC [Gretton et al., 2005]</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HSIC with RFFs* [Zhang et al., 2016]</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>FSIC (proposed)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* : RFFs = Random Fourier Features

- Focus on cases where $n$ (sample size) is large.
Goals

Want a test which is …

1. **Non-parametric** i.e., no parametric assumption on $P_{xy}$.
2. **Linear-time** i.e., computational complexity is $O(n)$. Fast.
3. **Adaptive** i.e., has a well-defined criterion for parameter tuning.

<table>
<thead>
<tr>
<th></th>
<th>Non-parametric</th>
<th>$O(n)$</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson correlation</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HSIC [Gretton et al., 2005]</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HSIC with RFFs* [Zhang et al., 2016]</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>FSIC (proposed)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

*: RFFs = Random Fourier Features

**Focus on cases where $n$ (sample size) is large.**
Witness Function [Gretton et al., 2012]

- A function showing the differences of two distributions $P$ and $Q$.
- Gaussian kernel: $k(x, v) = \exp\left(-\frac{||x-v||^2}{2\sigma^2}\right)$
- Empirical mean embedding of $P$: $\hat{\mu}_P(v) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)$
- Maximum Mean Discrepancy (MMD): $\|\hat{u}\|_{\text{RKHS}}$.
  - $\text{MMD}(P, Q) = 0$ if and only if $P = Q$. 
Witness Function [Gretton et al., 2012]

- A function showing the differences of two distributions $P$ and $Q$.
- Gaussian kernel: $k(x, v) = \exp\left(-\frac{||x-v||^2}{2\sigma^2}\right)$
- Empirical mean embedding of $P$: $\hat{\mu}_P(v) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)$
- Maximum Mean Discrepancy (MMD): $||\hat{u}||_{\text{RKHS}}$.
  - $\text{MMD}(P, Q) = 0$ if and only if $P = Q$.

\[\text{Observe } X = \{x_1, \ldots, x_n\} \sim P\]

\[\text{Observe } Y = \{y_1, \ldots, y_n\} \sim Q\]
Witness Function [Gretton et al., 2012]

- A function showing the differences of two distributions $P$ and $Q$.
- Gaussian kernel: $k(x, v) = \exp\left(-\frac{||x-v||^2}{2\sigma^2}\right)$
- Empirical mean embedding of $P$: $\hat{\mu}_P(v) = \frac{1}{n} \sum_{i=1}^n k(x_i, v)$
- Maximum Mean Discrepancy (MMD): $||\hat{u}||_{RKHS}$.
  - $\text{MMD}(P, Q) = 0$ if and only if $P = Q$. 

![Gaussian kernel illustration]

Gaussian kernel on $x_i$  

Gaussian kernel on $y_i$
Witness Function [Gretton et al., 2012]

- A function showing the differences of two distributions $P$ and $Q$.
- Gaussian kernel: $k(x, v) = \exp\left(-\frac{||x-v||^2}{2\sigma^2}\right)$
- Empirical mean embedding of $P$: $\hat{\mu}_P(v) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)$
- Maximum Mean Discrepancy (MMD): $\|\hat{u}\|_{\text{RKHS}}$.
  - $\text{MMD}(P, Q) = 0$ if and only if $P = Q$. 

\[ \hat{\mu}_P(v): \text{mean embedding of } P \]

\[ \hat{\mu}_Q(v): \text{mean embedding of } Q \]
Witness Function [Gretton et al., 2012]

- A function showing the differences of two distributions $P$ and $Q$.
- Gaussian kernel: $k(x, v) = \exp\left(-\frac{||x-v||^2}{2\sigma^2}\right)$
- Empirical mean embedding of $P$: $\hat{\mu}_P(v) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)$
- Maximum Mean Discrepancy (MMD): $||\hat{u}||_{\mathcal{RKHS}}$.
  - $\text{MMD}(P, Q) = 0$ if and only if $P = Q$. 

\[ \hat{u}(v) = \text{witness}(v) = \hat{\mu}_P(v) - \hat{\mu}_Q(v) \]
Witness Function [Gretton et al., 2012]

- A function showing the differences of two distributions $P$ and $Q$.
- Gaussian kernel: $k(x, v) = \exp \left( -\frac{\|x-v\|^2}{2\sigma^2} \right)$
- Empirical mean embedding of $P$: $\hat{\mu}_P(v) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)$
- Maximum Mean Discrepancy (MMD): $\| \hat{u} \|_{\text{RKHS}}$.
  - $\text{MMD}(P, Q) = 0$ if and only if $P = Q$.
Independence Test with HSIC [Gretton et al., 2005]

- Hilbert-Schmidt Independence Criterion.

\[
\text{HSIC}(X, Y) = \text{MMD}(P_{xy}, P_x P_y) = \|u\|_{\text{RKHS}}
\]

(need two kernels: \(k\) for \(X\), and \(l\) for \(Y\)).

- Empirical witness:

\[
\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w)
\]

where \(\hat{\mu}_{xy}(v, w) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)l(y_i, w)\).

- \(\text{HSIC}(X, Y) = 0\) if and only if \(X\) and \(Y\) are independent.
- Test statistic = \(\|\hat{u}\|_{\text{RKHS}}\) (“flatness” of \(\hat{u}\)). Complexity: \(\mathcal{O}(n^2)\).

Key: Can we measure the flatness by other way that costs only \(\mathcal{O}(n)\)?
Independence Test with HSIC [Gretton et al., 2005]

- **Hilbert-Schmidt Independence Criterion.**

\[
\text{HSIC}(X, Y) = \text{MMD}(P_{xy}, P_x P_y) = \|u\|_{\text{RKHS}}
\]

(need two kernels: \(k\) for \(X\), and \(l\) for \(Y\)).

- **Empirical witness:**

\[
\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w)
\]

where \(\hat{\mu}_{xy}(v, w) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)l(y_i, w)\).

- \(\text{HSIC}(X, Y) = 0\) if and only if \(X\) and \(Y\) are independent.
- Test statistic = \(\|\hat{u}\|_{\text{RKHS}}\) (“flatness” of \(\hat{u}\)). Complexity: \(\mathcal{O}(n^2)\).

**Key:** Can we measure the flatness by other way that costs only \(\mathcal{O}(n)\)?
Independence Test with HSIC [Gretton et al., 2005]

- **Hilbert-Schmidt Independence Criterion.**

\[
\text{HSIC}(X, Y) = \text{MMD}(P_{X,Y}, P_X P_Y) = \|u\|_{\text{RKHS}}
\]

(need two kernels: \(k\) for \(X\), and \(l\) for \(Y\)).

- **Empirical witness:**

\[
\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w)
\]

where \(\hat{\mu}_{xy}(v, w) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)l(y_i, w)\).

HSIC\((X, Y) = 0\) if and only if \(X\) and \(Y\) are independent.

Test statistic = \(\|\hat{u}\|_{\text{RKHS}}\) ("flatness" of \(\hat{u}\)). Complexity: \(\mathcal{O}(n^2)\).

**Key:** Can we measure the flatness by other way that costs only \(\mathcal{O}(n)\)?
Independence Test with HSIC [Gretton et al., 2005]

- **Hilbert-Schmidt Independence Criterion.**

\[
\text{HSIC}(X, Y) = \text{MMD}(P_{xy}, P_x P_y) = \|u\|_{\text{RKHS}}
\]

(need two kernels: \(k\) for \(X\), and \(l\) for \(Y\)).

- **Empirical witness:**

\[
\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w)
\]

where \(\hat{\mu}_{xy}(v, w) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)l(y_i, w)\).

\[
\hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w) = \text{Witness } \hat{u}(v, w)
\]

- **HSIC**\((X, Y) = 0\) if and only if \(X\) and \(Y\) are independent.
- **Test statistic** = \(\|\hat{u}\|_{\text{RKHS}}\) ("flatness" of \(\hat{u}\)). Complexity: \(O(n^2)\).

**Key:** Can we measure the flatness by other way that costs only \(O(n)\)?
Independence Test with HSIC [Gretton et al., 2005]

- Hilbert-Schmidt Independence Criterion.

\[
\text{HSIC}(X, Y) = \text{MMD}(P_{xy}, P_x P_y) = \|u\|_{\text{RKHS}}
\]

(need two kernels: \(k\) for \(X\), and \(l\) for \(Y\)).

- Empirical witness:

\[
\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w)
\]

where \(\hat{\mu}_{xy}(v, w) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)l(y_i, w)\).

\[
\hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w) = \text{Witness } \hat{u}(v, w)
\]

- \(\text{HSIC}(X, Y) = 0\) if and only if \(X\) and \(Y\) are independent.

- Test statistic = \(\|\hat{u}\|_{\text{RKHS}}\) (“flatness” of \(\hat{u}\)). Complexity: \(\mathcal{O}(n^2)\).

Key: Can we measure the flatness by other way that costs only \(\mathcal{O}(n)\)?
Independence Test with HSIC [Gretton et al., 2005]

- **Hilbert-Schmidt Independence Criterion.**

\[
\text{HSIC}(X, Y) = \text{MMD}(P_{xy}, P_x P_y) = \|u\|_{\text{RKHS}}
\]

(need two kernels: \(k\) for \(X\), and \(l\) for \(Y\)).

- **Empirical witness:**

\[
\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w)
\]

where \(\hat{\mu}_{xy}(v, w) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)l(y_i, w)\).

- **HSIC**\((X, Y) = 0\) if and only if \(X\) and \(Y\) are independent.
- **Test statistic** = \(\|\hat{u}\|_{\text{RKHS}}\) (“flatness” of \(\hat{u}\)). Complexity: \(\mathcal{O}(n^2)\).

**Key:** Can we measure the flatness by other way that costs only \(\mathcal{O}(n)\)?
Independence Test with HSIC [Gretton et al., 2005]

- **Hilbert-Schmidt Independence Criterion.**

  \[
  \text{HSIC}(X, Y) = \text{MMD}(P_{xy}, P_x P_y) = \|u\|_{\text{RKHS}}
  \]

  (need two kernels: \(k\) for \(X\), and \(l\) for \(Y\)).

- **Empirical witness:**

  \[
  \hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w)
  \]

  where \(\hat{\mu}_{xy}(v, w) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, v)l(y_i, w)\).

- **Key:** Can we measure the flatness by other way that costs only \(O(n)\)?
Proposal: The Finite Set Independence Criterion (FSIC)

**Idea:** Evaluate $\hat{u}^2(v, w)$ at only finitely many test locations.

- A set of random $J$ locations: $\{(v_1, w_1), \ldots, (v_J, w_J)\}$
- $$\text{FSIC}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i)$$

**Complexity:** $\mathcal{O}((d_x + d_y)Jn)$. Linear time.

**But**, what about an unlucky set of locations??
- Can $\text{FSIC}^2(X, Y) = 0$ even if $X$ and $Y$ are dependent??
**Proposal: The Finite Set Independence Criterion (FSIC)**

**Idea:** Evaluate $\hat{u}^2(v, w)$ at only finitely many test locations.

- A set of random $J$ locations: $\{(v_1, w_1), \ldots, (v_J, w_J)\}$
- $\hat{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i)$

Complexity: $\mathcal{O}((d_x + d_y)Jn)$. Linear time.

But, what about an unlucky set of locations??
- Can $\text{FSIC}^2(X, Y) = 0$ even if $X$ and $Y$ are dependent??
Proposal: The Finite Set Independence Criterion (FSIC)

**Idea:** Evaluate \( \hat{u}^2(v, w) \) at only finitely many test locations.

- A set of random \( J \) locations: \( \{(v_1, w_1), \ldots, (v_J, w_J)\} \)
- \( \widehat{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i) \)

![Diagram with a 2D distribution and several test locations marked with red stars.]

- Complexity: \( \mathcal{O}( (d_x + d_y) Jn ) \). Linear time.

**But, what about an unlucky set of locations??**
  - Can \( \text{FSIC}^2(X, Y) = 0 \) even if \( X \) and \( Y \) are dependent??
Proposal: The Finite Set Independence Criterion (FSIC)

**Idea:** Evaluate $\hat{u}^2(v, w)$ at only finitely many test locations.

- A set of random $J$ locations: $\{(v_1, w_1), \ldots, (v_J, w_J)\}$
- $\widehat{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i)$

![Image of a contour plot with red stars indicating test locations.]

- Complexity: $\mathcal{O}((d_x + d_y)Jn)$. Linear time.
- **But,** what about an unlucky set of locations??
  - Can $\text{FSIC}^2(X, Y) = 0$ even if $X$ and $Y$ are dependent??
Proposal: The Finite Set Independence Criterion (FSIC)

**Idea:** Evaluate $\hat{u}^2(v, w)$ at only finitely many test locations.

- A set of random $J$ locations: $\{(v_1, w_1), \ldots, (v_J, w_J)\}$
- $\overline{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i)$

![Heatmap with red stars indicating test locations and color bar ranging from 0.000 to 0.024]

- Complexity: $O((d_x + d_y)Jn)$. Linear time.
- **But**, what about an unlucky set of locations??
  - Can $\text{FSIC}^2(X, Y) = 0$ even if $X$ and $Y$ are dependent??
Proposal: The Finite Set Independence Criterion (FSIC)

**Idea:** Evaluate $\hat{u}^2(v, w)$ at only finitely many test locations.

- A set of random $J$ locations: $\{(v_1, w_1), \ldots, (v_J, w_J)\}$
- $\text{FSIC}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i)$

![Image of a 2D density plot with red stars indicating test locations.](image)

- Complexity: $\mathcal{O}((d_x + d_y) Jn)$. Linear time.

**But**, what about an unlucky set of locations??

- Can $\text{FSIC}^2(X, Y) = 0$ even if $X$ and $Y$ are dependent??

- No. Population $\text{FSIC}(X, Y) = 0$ iff $X \perp Y$, almost surely.
Requirements on the Kernels

**Definition 1 (Analytic kernels).**

\( k : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \) is said to be **analytic** if for all \( x \in \mathcal{X}, \ v \to k(x, v) \) is a real analytic function on \( \mathcal{X} \).

- Analytic: Taylor series about \( x_0 \) converges for all \( x_0 \in \mathcal{X} \).
- \( \implies \) \( k \) is infinitely differentiable.

**Definition 2 (Characteristic kernels).**

- Let \( P, Q \) be two distributions, and \( g \) be a kernel.
- Let \( \mu_P(v) := \mathbb{E}_{z \sim P}[g(z, v)] \) and \( \mu_Q(v) := \mathbb{E}_{z \sim Q}[g(z, v)] \).

\( g \) is said to be **characteristic** if \( P \neq Q \) implies \( \mu_P \neq \mu_Q \).
Requirements on the Kernels

Definition 1 (Analytic kernels).

$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be analytic if for all $x \in \mathcal{X}$, $v \to k(x, v)$ is a real analytic function on $\mathcal{X}$.

- Analytic: Taylor series about $x_0$ converges for all $x_0 \in \mathcal{X}$.
- $\implies k$ is infinitely differentiable.

Definition 2 (Characteristic kernels).

- Let $P, Q$ be two distributions, and $g$ be a kernel.
- Let $\mu_P(v) := \mathbb{E}_{z \sim P}[g(z, v)]$ and $\mu_Q(v) := \mathbb{E}_{z \sim Q}[g(z, v)]$.

$g$ is said to be characteristic if $P \neq Q$ implies $\mu_P \neq \mu_Q$.
Proposition 1.

Assume

1. The product kernel \( g((x, y), (x', y')) := k(x, x')l(y, y') \) is characteristic and analytic (i.e., \( k, l \) are Gaussian kernels).

2. Test locations \( \{(v_i, w_i)\}_{i=1}^J \sim \eta \) where \( \eta \) has a density.

Then, \( \eta \)-almost surely, \( \text{FSIC}(X, Y) = 0 \) iff \( X \) and \( Y \) are independent.
**Proposition 1.**

Assume

1. The product kernel $g((x, y), (x', y')) := k(x, x')l(y, y')$ is characteristic and analytic (i.e., $k, l$ are Gaussian kernels).

2. Test locations $\{(v_i, w_i)\}_{i=1}^J \sim \eta$ where $\eta$ has a density.

Then, $\eta$-almost surely, $\text{FSIC}(X, Y) = 0$ iff $X$ and $Y$ are independent.

Let’s plot
Proposition 1. Assume

1. The product kernel \( g((x, y), (x', y')) := k(x, x')l(y, y') \) is characteristic and analytic (i.e., \( k, l \) are Gaussian kernels).

2. Test locations \( \{(v_i, w_i)\}_{i=1}^J \sim \eta \) where \( \eta \) has a density.

Then, \( \eta \)-almost surely, \( \text{FSIC}(X, Y) = 0 \) iff \( X \) and \( Y \) are independent.
Proposition 1.

Assume

1. The product kernel \( g((x, y), (x', y')) := k(x, x')l(y, y') \) is characteristic and analytic (i.e., \( k, l \) are Gaussian kernels).

2. Test locations \( \{(v_i, w_i)\}^J_{i=1} \sim \eta \) where \( \eta \) has a density.

Then, \( \eta \)-almost surely, \( \text{FSIC}(X, Y) = 0 \) iff \( X \) and \( Y \) are independent.
**Proposition 1.**

Assume

1. The product kernel \( g((x, y), (x', y')) := k(x, x')l(y, y') \) is characteristic and analytic (i.e., \( k, l \) are Gaussian kernels).

2. Test locations \( \{(v_i, w_i)\}_{i=1}^{J} \sim \eta \) where \( \eta \) has a density.

Then, \( \eta \)-almost surely, \( \text{FSIC}(X, Y) = 0 \) iff \( X \) and \( Y \) are independent.

Under \( H_1 \), \( u \) is not a zero function (\( P \mapsto \mathbb{E}_{z \sim P}[g(z, \cdot)] \) is injective).

\( u \) is analytic. So, \( R_u = \{(v, w) \mid u(v, w) = 0\} \) has 0 Lebesgue measure.

So, \( \{(v_i, w_i)\}_{i=1}^{J} \sim \eta \) will not be in \( R_u \) (with probability 1).
Alternative View of the Witness $u(v, w)$

The witness $u(v, w)$ can be rewritten as

$$u(v, w) := \mu_{xy}(v, w) - \mu_x(v)\mu_y(w)$$

$$= \mathbb{E}_{xy}[k(x, v)l(y, w)] - \mathbb{E}_x[k(x, v)]\mathbb{E}_y[l(y, w)],$$

$$= \text{cov}_{xy}[k(x, v), l(y, w)].$$

1. Transforming $x \mapsto k(x, v)$ and $y \mapsto l(y, w)$ (from $\mathbb{R}^{d_y}$ to $\mathbb{R}$).
2. Then, take the covariance.

The kernel transformations turn the linear covariance into a dependence measure.
Alternative View of the Witness $u(v, w)$

The witness $u(v, w)$ can be rewritten as

$$
u(v, w) := \mu_{xy}(v, w) - \mu_x(v)\mu_y(w)
= \mathbb{E}_{xy}[k(x, v)l(y, w)] - \mathbb{E}_x[k(x, v)]\mathbb{E}_y[l(y, w)],
= \text{cov}_{xy}[k(x, v), l(y, w)].$$

1. Transforming $x \mapsto k(x, v)$ and $y \mapsto l(y, w)$ (from $\mathbb{R}^{d_y}$ to $\mathbb{R}$).
2. Then, take the covariance.

The kernel transformations turn the linear covariance into a dependence measure.
Alternative Form of $\hat{u}(v, w)$

- Recall $\widehat{\text{FSIC}}^2 = \frac{1}{J} \sum_{i=1}^{J} \hat{u}(v_i, w_i)^2$
- Let $\hat{\mu}_x \hat{\mu}_y(v, w)$ be an unbiased estimator of $\mu_x(v) \mu_y(w)$.
- $\hat{\mu}_x \hat{\mu}_y(v, w) := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} k(x_i, v)l(y_j, w)$.
- An unbiased estimator of $u(v, w)$ is

$$\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x \hat{\mu}_y(v, w)$$

$$= \frac{2}{n(n-1)} \sum_{i<j} h_{(v, w)}((x_i, y_i), (x_j, y_j)),$$

where

$$h_{(v, w)}((x, y), (x', y')) := \frac{1}{2}(k(x, v) - k(x', v))(l(y, w) - l(y', w)).$$

- For a fixed $(v, w)$, $\hat{u}(v, w)$ is a one-sample 2nd-order U-statistic.
Alternative Form of $\hat{u}(v, w)$

- Recall $\hat{\text{FSIC}}^2 = \frac{1}{J} \sum_{i=1}^{J} \hat{u}(v_i, w_i)^2$
- Let $\hat{\mu}_x \hat{\mu}_y (v, w)$ be an unbiased estimator of $\mu_x(v) \mu_y(w)$.
- $\hat{\mu}_x \hat{\mu}_y (v, w) := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} k(x_i, v) l(y_j, w)$.
- An unbiased estimator of $u(v, w)$ is

$$\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x \hat{\mu}_y(v, w)$$

$$= \frac{2}{n(n-1)} \sum_{i<j} h_{(v, w)}((x_i, y_i), (x_j, y_j))$$

where

$$h_{(v, w)}((x, y), (x', y')) := \frac{1}{2} (k(x, v) - k(x', v))(l(y, w) - l(y', w)).$$

- For a fixed $(v, w)$, $\hat{u}(v, w)$ is a one-sample 2nd-order U-statistic.
Alternative Form of $\hat{u}(v, w)$

- Recall $\text{FSIC}^2 = \frac{1}{J} \sum_{i=1}^{J} \hat{u}(v_i, w_i)^2$
- Let $\hat{\mu}_x \hat{\mu}_y(v, w)$ be an unbiased estimator of $\mu_x(v) \mu_y(w)$.
- $\hat{\mu}_x \hat{\mu}_y(v, w) := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} k(x_i, v) l(y_j, w)$.
- An unbiased estimator of $u(v, w)$ is

$$\hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x \hat{\mu}_y(v, w)$$

$$= \frac{2}{n(n-1)} \sum_{i<j} h_{(v,w)}((x_i, y_i), (x_j, y_j)),$$

where

$$h_{(v,w)}((x, y), (x', y')) := \frac{1}{2}(k(x, v) - k(x', v))(l(y, w) - l(y', w)).$$

- For a fixed $(v, w)$, $\hat{u}(v, w)$ is a one-sample 2nd-order U-statistic.
Asymptotic Distribution of \( \hat{u} \)

\[
\hat{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i) = \frac{1}{J} \hat{u}^\top \hat{u},
\]

where \( \hat{u} = (\hat{u}(v_1, w_1), \ldots, \hat{u}(v_J, w_J))^\top \).

**Proposition 2 (Asymptotic distribution of \( \hat{u} \)).**

For any fixed locations \( \{(v_i, w_i)\}_{i=1}^{J} \), we have \( \sqrt{n}(\hat{u} - u) \xrightarrow{d} \mathcal{N}(0, \Sigma) \).

- \( \Sigma_{ij} = \mathbb{E}_{xy}[\tilde{k}(x, v_i)\tilde{l}(y, w_i)\tilde{k}(x, v_j)\tilde{l}(y, w_j)] - u(v_i, w_i)u(v_j, w_j) \),
- \( \tilde{k}(x, v) := k(x, v) - \mathbb{E}_{x'}k(x', v) \),
- \( \tilde{l}(y, w) := l(y, w) - \mathbb{E}_{y'}l(y', w) \).

Under \( H_0 \),

\[
n\hat{\text{FSIC}}^2 = \frac{n}{J} \hat{u}^\top \hat{u} \sim \text{weighted sum of dependent } \chi^2 \text{ variables}.
\]

- **Difficult** to get \((1 - \alpha)\)-quantile for the threshold.
Asymptotic Distribution of $\hat{u}$

$$\overline{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i) = \frac{1}{J} \hat{u}^\top \hat{u},$$

where $\hat{u} = (\hat{u}(v_1, w_1), \ldots, \hat{u}(v_J, w_J))^\top$.

**Proposition 2 (Asymptotic distribution of $\hat{u}$).**

For any fixed locations $\{(v_i, w_i)\}_{i=1}^{J}$, we have $\sqrt{n}(\hat{u} - u) \xrightarrow{d} \mathcal{N}(0, \Sigma)$.

- $\Sigma_{ij} = \mathbb{E}_{xy}[\tilde{k}(x, v_i)\tilde{l}(y, w_i)\tilde{k}(x, v_j)\tilde{l}(y, w_j)] - u(v_i, w_i)u(v_j, w_j)$,
- $\tilde{k}(x, v) := k(x, v) - \mathbb{E}_{x'} k(x', v)$,
- $\tilde{l}(y, w) := l(y, w) - \mathbb{E}_{y'} l(y', w)$.

Under $H_0$, $n\overline{\text{FSIC}}^2 = \frac{n}{J} \hat{u}^\top \hat{u} \sim$ weighted sum of dependent $\chi^2$ variables.

- **Difficult** to get $(1 - \alpha)$-quantile for the threshold.
Asymptotic Distribution of $\hat{u}$

$$\tilde{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i) = \frac{1}{J} \hat{u}^\top \hat{u},$$

where $\hat{u} = (\hat{u}(v_1, w_1), \ldots, \hat{u}(v_J, w_J))^\top$.

**Proposition 2 (Asymptotic distribution of $\hat{u}$).**

For any fixed locations $\{(v_i, w_i)\}_{i=1}^{J}$, we have $\sqrt{n}(\hat{u} - u) \xrightarrow{d} \mathcal{N}(0, \Sigma)$.

- $\Sigma_{ij} = \mathbb{E}_{xy}[^{\tilde{k}}(x, v_i)^{\tilde{l}}(y, w_i) ^{\tilde{k}}(x, v_j)^{\tilde{l}}(y, w_j)] - u(v_i, w_i)u(v_j, w_j),$
- $^{\tilde{k}}(x, v) := k(x, v) - \mathbb{E}_{x'}k(x', v),$
- $^{\tilde{l}}(y, w) := l(y, w) - \mathbb{E}_{y'}l(y', w)$.

Under $H_0$,

$$n\tilde{\text{FSIC}}^2 = \frac{n}{J} \hat{u}^\top \hat{u} \sim \text{weighted sum of dependent } \chi^2 \text{ variables}.$$

- Difficult to get $(1 - \alpha)$-quantile for the threshold.
Asymptotic Distribution of $\hat{u}$

$$\overline{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i) = \frac{1}{J} \hat{u}^\top \hat{u},$$

where $\hat{u} = (\hat{u}(v_1, w_1), \ldots, \hat{u}(v_J, w_J))^\top$.

Proposition 2 (Asymptotic distribution of $\hat{u}$).

For any fixed locations $\{(v_i, w_i)\}_{i=1}^{J}$, we have $\sqrt{n}(\hat{u} - u) \overset{d}{\to} \mathcal{N}(0, \Sigma)$.

- $\Sigma_{ij} = \mathbb{E}_{xy}[\tilde{k}(x, v_i)\tilde{l}(y, w_i)\tilde{k}(x, v_j)\tilde{l}(y, w_j)] - u(v_i, w_i)u(v_j, w_j)$,
- $\tilde{k}(x, v) := k(x, v) - \mathbb{E}_{x'}k(x', v)$,
- $\tilde{l}(y, w) := l(y, w) - \mathbb{E}_{y'}l(y', w)$.

Under $H_0$,

$$n\overline{\text{FSIC}}^2 = \frac{n}{J} \hat{u}^\top \hat{u} \sim \text{weighted sum of dependent } \chi^2 \text{ variables}.$$
Asymptotic Distribution of $\hat{u}$

$$\widehat{\text{FSIC}}^2(X, Y) = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i) = \frac{1}{J} \hat{u}^\top \hat{u},$$

where $\hat{u} = (\hat{u}(v_1, w_1), \ldots, \hat{u}(v_J, w_J))^\top$.

**Proposition 2 (Asymptotic distribution of $\hat{u}$).**

For any fixed locations $\{(v_i, w_i)\}_{i=1}^{J}$, we have $\sqrt{n}(\hat{u} - u) \xrightarrow{d} \mathcal{N}(0, \Sigma)$.

- $\Sigma_{ij} = \mathbb{E}_{xy}[\tilde{k}(x, v_i)\tilde{l}(y, w_i)\tilde{k}(x, v_j)\tilde{l}(y, w_j)] - u(v_i, w_i)u(v_j, w_j)$,
- $\tilde{k}(x, v) := k(x, v) - \mathbb{E}_{x'}k(x', v)$,
- $\tilde{l}(y, w) := l(y, w) - \mathbb{E}_{y'}l(y', w)$.

Under $H_0$,

$$n\widehat{\text{FSIC}}^2 = \frac{n}{J} \hat{u}^\top \hat{u} \sim \text{weighted sum of dependent } \chi^2 \text{ variables.}$$

- **Difficult** to get $(1 - \alpha)$-quantile for the threshold.
Normalized FSIC (NFSIC)

\[
\text{NFSIC}^2(X, Y) = \hat{\lambda}_n := n \hat{u}^\top (\hat{\Sigma} + \gamma_n \mathbf{I})^{-1} \hat{u},
\]

with a regularization parameter \( \gamma_n \geq 0 \).

- **Key**: NFSIC = FSIC normalized by the covariance.

### Theorem 1 (NFSIC test is consistent).

Assume

1. The product kernel is characteristic and analytic.
2. \( \lim_{n \to \infty} \gamma_n = 0 \).

Then, for any \( k, l \) and \( \{(v_i, w_i)\}_{i=1}^J \sim \eta \),

1. Under \( H_0 \), \( \hat{\lambda}_n \overset{d}{\to} \chi^2(J) \) as \( n \to \infty \).
2. Under \( H_1 \), \( \lim_{n \to \infty} \mathbb{P} \left( \hat{\lambda}_n \geq T_\alpha \right) = 1 \), \( \eta \)-almost surely.
Normalized FSIC (NFSIC)

\[ \overline{\text{NFSIC}}^2(X, Y) = \hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{u}, \]

with a regularization parameter \( \gamma_n \geq 0 \).

- **Key**: NFSIC = FSIC normalized by the covariance.

### Theorem 1 (NFSIC test is consistent).

Assume

1. The product kernel is characteristic and analytic.
2. \( \lim_{n \to \infty} \gamma_n = 0 \).

Then, for any \( k, l \) and \( \{(v_i, w_i)\}_{i=1}^J \sim \eta \),

1. Under \( H_0 \), \( \hat{\lambda}_n \xrightarrow{\text{d}} \chi^2(J) \) as \( n \to \infty \).
2. Under \( H_1 \), \( \lim_{n \to \infty} \mathbb{P} \left( \hat{\lambda}_n \geq T_\alpha \right) = 1 \), \( \eta \)-almost surely.
**Normalized FSIC (NFSIC)**

\[
\text{NFSIC}^2(X, Y) = \hat{\lambda}_n := n\hat{\mu}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{\mu},
\]

with a regularization parameter \(\gamma_n \geq 0\).

■ **Key**: NFSIC = FSIC normalized by the covariance.

---

**Theorem 1 (NFSIC test is consistent).**

*Assume*

1. The product kernel is characteristic and analytic.
2. \(\lim_{n \to \infty} \gamma_n = 0\).

*Then, for any \(k, l\) and \(\{(v_i, w_i)\}_{i=1}^J \sim \eta,\)*

1. Under \(H_0\), \(\hat{\lambda}_n \xrightarrow{d} \chi^2(J)\) as \(n \to \infty\).
2. Under \(H_1\), \(\lim_{n \to \infty} \mathbb{P}(\hat{\lambda}_n \geq T_\alpha) = 1, \eta\)-almost surely.*
Normalized FSIC (NFSIC)

\[
\text{NFSIC}^2(X, Y) = \hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{u},
\]

with a regularization parameter \(\gamma_n \geq 0\).

**Key**: NFSIC = FSIC normalized by the covariance.

**Theorem 1 (NFSIC test is consistent).**

Assume

1. The product kernel is characteristic and analytic.
2. \(\lim_{n \to \infty} \gamma_n = 0\).

Then, for any \(k, l\) and \(\{(v_i, w_i)\}_{i=1}^J \sim \eta\),

1. Under \(H_0\), \(\hat{\lambda}_n \xrightarrow{d} \chi^2(J)\) as \(n \to \infty\).
2. Under \(H_1\), \(\lim_{n \to \infty} \mathbb{P}\left(\hat{\lambda}_n \geq T_\alpha\right) = 1\), \(\eta\)-almost surely.
Normalized FSIC (NFSIC)

\[ \text{NFSIC}^2(X, Y) = \hat{\lambda}_n := n\hat{\mu}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{\mu}, \]

with a regularization parameter \( \gamma_n \geq 0. \)

- **Key**: NFSIC = FSIC normalized by the covariance.

**Theorem 1 (NFSIC test is consistent).**

**Assume**

1. **The product kernel is characteristic and analytic.**
2. \( \lim_{n \to \infty} \gamma_n = 0. \)

**Then,** for any \( k, l \) and \( \{(v_i, w_i)\}_{i=1}^J \sim \eta, \)

1. **Under** \( H_0, \hat{\lambda}_n \xrightarrow{d} \chi^2(J) \) as \( n \to \infty. \)
2. **Under** \( H_1, \lim_{n \to \infty} \mathbb{P} \left( \hat{\lambda}_n \geq T_\alpha \right) = 1, \eta\text{-almost surely.} \)
Normalized FSIC (NFSIC)

\[
\widehat{\text{NFSIC}}^2(X, Y) = \lambda_n := n\hat{u}^\top \left(\hat{\Sigma} + \gamma_n \mathbf{I}\right)^{-1} \hat{u},
\]

with a regularization parameter \(\gamma_n \geq 0\).

Key: NFSIC = FSIC normalized by the covariance.

Theorem 1 (NFSIC test is consistent).

Assume

1. The product kernel is characteristic and analytic.
2. \(\lim_{n \to \infty} \gamma_n = 0\).

Then, for any \(k, l\) and \(\{(v_i, w_i)\}_{i=1}^J \sim \eta\),

1. Under \(H_0\), \(\lambda_n \overset{d}{\to} \chi^2(J)\) as \(n \to \infty\).
2. Under \(H_1\), \(\lim_{n \to \infty} \mathbb{P} \left(\lambda_n \geq T_\alpha\right) = 1\), \(\eta\)-almost surely.

Asymptotically, false positive rate is at \(\alpha\) under \(H_0\), and always reject under \(H_1\).
An Estimator of NFSIC$^2$

$$\hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{u},$$

- Test locations $\{(v_i, w_i)\}_{i=1}^J \sim \eta$.
- $K = [k(v_i, x_j)] \in \mathbb{R}^{J \times n}$
- $L = [l(w_i, y_j)] \in \mathbb{R}^{J \times n}$. (No $n \times n$ Gram matrix.)

Estimators

1. $\hat{u} = \frac{(K \circ L) 1_n}{n-1} - \frac{(K 1_n \circ (L 1_n))}{n(n-1)}$.
2. $\hat{\Sigma} = \frac{\Gamma \Gamma^\top}{n}$ where $\Gamma := (K - n^{-1}K1_n 1_n^\top) \circ (L - n^{-1}L1_n 1_n^\top) - \hat{u}1_n^\top$.

- $\hat{\lambda}_n$ can be computed in $O(J^3 + J^2 n + (d_x + d_y) Jn)$ time.

Main Point: Linear in $n$. Cubic in $J$ (small).
An Estimator of $\text{NFSIC}^2$

$$\hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{u},$$

- Test locations $\{(v_i, w_i)\}^J_{i=1} \sim \eta$.
- $K = [k(v_i, x_j)] \in \mathbb{R}^{J \times n}$
- $L = [l(w_i, y_j)] \in \mathbb{R}^{J \times n}$. (No $n \times n$ Gram matrix.)

**Estimators**

1. $\hat{u} = \frac{(K \circ L)1_n}{n-1} - \frac{(K1_n) \circ (L1_n)}{n(n-1)}$.

2. $\hat{\Sigma} = \frac{\Gamma \Gamma^\top}{n}$ where $\Gamma := (K - n^{-1}K1_n1_n^\top) \circ (L - n^{-1}L1_n1_n^\top) - \hat{u}1_n^\top$.

$\hat{\lambda}_n$ can be computed in $O(J^3 + J^2n + (d_x + d_y)Jn)$ time.

**Main Point:** Linear in $n$. Cubic in $J$ (small).
An Estimator of $\text{NFSIC}^2$

\[ \hat{\lambda}_n := n \hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{u}, \]

- Test locations $\{(v_i, w_i)\}_{i=1}^J \sim \eta$.
- $K = [k(v_i, x_j)] \in \mathbb{R}^{J \times n}$
- $L = [l(w_i, y_j)] \in \mathbb{R}^{J \times n}$. (No $n \times n$ Gram matrix.)

Estimators

1. $\hat{u} = \frac{(K \circ L)1_n}{n-1} - \frac{(K1_n) \circ (L1_n)}{n(n-1)}$.
2. $\hat{\Sigma} = \frac{\Gamma \Gamma^\top}{n}$ where $\Gamma := (K - n^{-1}K1_n1_n^\top) \circ (L - n^{-1}L1_n1_n^\top) - \hat{u}1_n^\top$.

- $\hat{\lambda}_n$ can be computed in $\mathcal{O}(J^3 + J^2n + (d_x + d_y)Jn)$ time.

Main Point: Linear in $n$. Cubic in $J$ (small).
An Estimator of $\text{NFSIC}^2$

\[ \hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n \mathbf{I})^{-1} \hat{u}, \]

- Test locations \( \{(v_i, w_i)\}_{i=1}^J \sim \eta. \)
- \( K = [k(v_i, x_j)] \in \mathbb{R}^{J \times n} \)
- \( L = [l(w_i, y_j)] \in \mathbb{R}^{J \times n}. \) (No \( n \times n \) Gram matrix.)

Estimators

1. \( \hat{u} = \frac{(K \circ L)1_n}{n-1} - \frac{(K1_n) \circ (L1_n)}{n(n-1)}. \)
2. \( \hat{\Sigma} = \frac{\Gamma \Gamma^\top}{n} \) where \( \Gamma := (K - n^{-1}K1_n1_n^\top) \circ (L - n^{-1}L1_n1_n^\top) - \hat{u}1_n^\top. \)

- \( \lambda_n \) can be computed in \( \mathcal{O}(J^3 + J^2 n + (d_x + d_y) J n) \) time.

**Main Point:** Linear in \( n. \) Cubic in \( J \) (small).
An Estimator of $\text{NFSIC}^2$

\[
\hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{u},
\]

- Test locations $\{(v_i, w_i)\}_{i=1}^J \sim \eta$.
- $K = [k(v_i, x_j)] \in \mathbb{R}^{J \times n}$
- $L = [l(w_i, y_j)] \in \mathbb{R}^{J \times n}$. (No $n \times n$ Gram matrix.)

**Estimators**

1. $\hat{u} = \frac{(K \circ L) 1_n}{n-1} - \frac{(K_{1n}) \circ (L_{1n})}{n(n-1)}$.

2. $\hat{\Sigma} = \frac{\Gamma \Gamma^\top}{n}$ where $\Gamma := (K - n^{-1}K_{1n}1_n^\top) \circ (L - n^{-1}L_{1n}1_n^\top) - \hat{u}1_n^\top$.

- $\hat{\lambda}_n$ can be computed in $O(J^3 + J^2n + (d_x + d_y)Jn)$ time.

**Main Point:** Linear in $n$. Cubic in $J$ (small).
Optimizing Test Locations $\{(v_i, w_i)\}_{i=1}^J$

- Test NFSIC$^2$ is consistent for any random locations $\{(v_i, w_i)\}_{i=1}^J$.
- In practice, tuning them will increase the test power.
Optimizing Test Locations $\{(v_i, w_i)\}_{i=1}^J$

- Test NFSIC$^2$ is consistent for any random locations $\{(v_i, w_i)\}_{i=1}^J$.
- In practice, tuning them will increase the test power.
Optimizing Test Locations $\{(v_i, w_i)\}_{i=1}^J$

- Test $\text{NFSIC}^2$ is consistent for any random locations $\{(v_i, w_i)\}_{i=1}^J$.
- In practice, tuning them will increase the test power.

Under $H_0 : X \perp Y$, we have $\hat{\lambda}_n \sim \chi^2(J)$ as $n \to \infty$. 

![Graph showing $\hat{\lambda}_n \sim \chi^2(J)$ as $n \to \infty$.]
Optimizing Test Locations \( \{ (v_i, w_i) \}_{i=1}^{J} \)

- Test \( \text{NFSIC}^2 \) is consistent for any random locations \( \{ (v_i, w_i) \}_{i=1}^{J} \).
- In practice, tuning them will increase the test power.

Under \( H_0 : X \perp Y \), we have \( \hat{\lambda}_n \sim \chi^2(J) \) as \( n \to \infty \).
Optimizing Test Locations \( \{(v_i, w_i)\}_{i=1}^{J} \)

- Test NFSIC\(^2\) is consistent for any random locations \( \{(v_i, w_i)\}_{i=1}^{J} \).
- In practice, tuning them will increase the test power.

Under \( H_1 \), \( \hat{\lambda}_n \) will be large. Follows some distribution \( \mathbb{P}_{H_1}(\hat{\lambda}_n) \)

![Diagram showing distributions](image)

- Blue: \( \chi^2(J) \)
- Green: \( T_\alpha \)
- Red: \( \mathbb{P}_{H_1}(\hat{\lambda}_n) \)
Optimizing Test Locations \( \{(v_i, w_i)\}_{i=1}^J \)

- Test NFSIC\(^2\) is consistent for any random locations \( \{(v_i, w_i)\}_{i=1}^J \).
- In practice, tuning them will increase the test power.

Test power = \( P(\text{reject } H_0 \mid H_1 \text{ true}) = P(\hat{\lambda}_n \geq T_\alpha) \)
Optimizing Test Locations $\{(v_i, w_i)\}_{i=1}^{J}$

- Test $\widehat{\text{NFSIC}}^2$ is consistent for any random locations $\{(v_i, w_i)\}_{i=1}^{J}$.
- In practice, tuning them will increase the test power.

Test power $= \mathbb{P}(\text{reject } H_0 \mid H_1 \text{ true}) = \mathbb{P}(\hat{\lambda}_n \geq T_\alpha)$

**Idea:** Pick locations and Gaussian widths to maximize (lower bound of) test power.
Optimizing Test Locations \( \{(v_i, w_i)\}_{i=1}^{J} \)

- Test \( \text{NFSIC}^2 \) is consistent for any random locations \( \{(v_i, w_i)\}_{i=1}^{J} \).
- In practice, tuning them will increase the test power.

Test power = \( \mathbb{P}(\text{reject } H_0 \mid H_1 \text{ true}) = \mathbb{P}(\hat{\lambda}_n \geq T_\alpha) \)

**Idea:** Pick locations and Gaussian widths to maximize (lower bound of) test power.
Recall $\hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{u}$.

Theorem 2 (A lower bound on the test power).

Let $\text{NFSIC}^2(X, Y) := \lambda_n := nu^\top \Sigma^{-1} u$.

With some conditions, for any $k, l$, and $\{(v_i, w_i)\}_{i=1}^J$, the test power satisfies $\mathbb{P}\left(\hat{\lambda}_n \geq T_\alpha\right) \geq L(\lambda_n)$ where

$$L(\lambda_n) = 1 - 62e^{-\xi_1 \gamma_n^2 (\lambda_n - T_\alpha)^2 / n} - 2e^{-\left[0.5n / (\lambda_n - T_\alpha)^2 / [\xi_2 n^2] \right]}$$

$$- 2e^{-\left[(\lambda_n - T_\alpha) \gamma_n (n-1) / 3 - \xi_3 n - c_3 \gamma_n^2 n(n-1)]^2 / [\xi_4 n^2 (n-1)]\right]}$$

where $\xi_1, \ldots, \xi_4, c_3 > 0$ are constants. For large $n$, $L(\lambda_n)$ is increasing in $\lambda_n$.

Do: Locations and Gaussian widths $= \arg\max L(\lambda_n) = \arg\max \lambda_n$
Optimization Objective = Power Lower Bound

- Recall $\hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1} \hat{u}$.

Theorem 2 (A lower bound on the test power).

- Let $\text{NFSIC}^2(X, Y) := \lambda_n := nu^\top \Sigma^{-1} u$.

With some conditions, for any $k, l$, and $\{(v_i, w_i)\}_{i=1}^J$, the test power satisfies $P(\hat{\lambda}_n \geq T_\alpha) \geq L(\lambda_n)$ where

$$L(\lambda_n) = 1 - 62e^{-\xi_1 \gamma_n^2 (\lambda_n - T_\alpha)^2/n} - 2e^{-[0.5n](\lambda_n - T_\alpha)^2/\xi_2 n^2} - 2e^{-[(\lambda_n - T_\alpha)\gamma_n(n-1)/3 - \xi_3 n - c_3 \gamma_n^2 n(n-1)]^2/\xi_4 n^2 (n-1)},$$

where $\xi_1, \ldots, \xi_4, c_3 > 0$ are constants. For large $n$, $L(\lambda_n)$ is increasing in $\lambda_n$.

Do: Locations and Gaussian widths = $\arg \max L(\lambda_n) = \arg \max \lambda_n$
Optimization Objective = Power Lower Bound

- Recall $\hat{\lambda}_n := n\hat{u}^\top \left( \hat{\Sigma} + \gamma_n I \right)^{-1} \hat{u}$.

**Theorem 2** (A lower bound on the test power).

- Let $\text{NFSIC}^2(X, Y) := \lambda_n := nu^\top \Sigma^{-1} u$.

With some conditions, for any $k, l$, and $\{(v_i, w_i)\}_{i=1}^J$, the test power satisfies $P\left( \hat{\lambda}_n \geq T_\alpha \right) \geq L(\lambda_n)$ where

$$L(\lambda_n) = 1 - 62e^{-\xi_1 \gamma_n^2 (\lambda_n - T_\alpha)^2 / n} - 2e^{-\left[0.5n(\lambda_n - T_\alpha)^2 / [\xi_2 n^2] \right]} - 2e^{-\left[ (\lambda_n - T_\alpha) \gamma_n (n-1) / 3 - \xi_3 n - c_3 \gamma_n^2 n(n-1) \right]^2 / [\xi_4 n^2 (n-1)]},$$

where $\xi_1, \ldots, \xi_4, c_3 > 0$ are constants. For large $n$, $L(\lambda_n)$ is increasing in $\lambda_n$.

**Do:** Locations and Gaussian widths $= \text{arg max } L(\lambda_n) = \text{arg max } \lambda_n$
Optimization Objective = Power Lower Bound

- Recall $\hat{\lambda}_n := n\hat{u}^\top (\hat{\Sigma} + \gamma_n I)^{-1}\hat{u}$.

**Theorem 2** (A lower bound on the test power).

- Let $\text{NFSIC}^2(X, Y) := \lambda_n := nu^\top \Sigma^{-1}u$.

With some conditions, for any $k, l, \text{ and } \{(v_i, w_i)\}_{i=1}^{J}$, the test power satisfies $\mathbb{P}(\hat{\lambda}_n \geq T_\alpha) \geq L(\lambda_n)$ where

$$L(\lambda_n) = 1 - 62e^{-\xi_1\gamma_n^2(\lambda_n - T_\alpha)^2/n} - 2e^{-[0.5n](\lambda_n - T_\alpha)^2/[\xi_2 n^2]} - 2e^{-[(\lambda_n - T_\alpha)\gamma_n(n-1)/3 - \xi_3 n - c_3 \gamma_n^2 n(n-1)]^2/[\xi_4 n^2(n-1)]},$$

where $\xi_1, \ldots, \xi_4, c_3 > 0$ are constants. For large $n$, $L(\lambda_n)$ is increasing in $\lambda_n$.

**Do:** Locations and Gaussian widths $= \arg \max L(\lambda_n) = \arg \max \lambda_n$
Optimization Procedure

- \( \text{NFSIC}^2(X, Y) := \lambda_n := nu^\top \Sigma^{-1} u \) is unknown.
- Split the data into 2 disjoint sets: training (tr) and test (te) sets.

Procedure:

1. Estimate \( \lambda_n \) with \( \hat{\lambda}_n^{(\text{tr})} \) (i.e., computed on the training set).
2. Optimize all \( \{(v_i, w_i)\}_{i=1}^{J} \) and Gaussian widths with gradient ascent.
3. Independence test with \( \hat{\lambda}_n^{(\text{te})} \). Reject \( H_0 \) if \( \hat{\lambda}_n^{(\text{te})} > T_\alpha \).

- Splitting avoids overfitting.
Optimization Procedure

- $\text{NFSIC}^2(X, Y) := \lambda_n := n u^\top \Sigma^{-1} u$ is unknown.
- Split the data into 2 disjoint sets: training (tr) and test (te) sets.

**Procedure:**

1. Estimate $\lambda_n$ with $\hat{\lambda}_n^{(tr)}$ (i.e., computed on the training set).
2. Optimize all $\{(v_i, w_i)\}_{i=1}^J$ and Gaussian widths with gradient ascent.
3. Independence test with $\hat{\lambda}_n^{(te)}$. Reject $H_0$ if $\hat{\lambda}_n^{(te)} \geq T_\alpha$.

- Splitting avoids overfitting.
Optimization Procedure

\[ \text{NFSIC}^2(X, Y) := \lambda_n := n u^\top \Sigma^{-1} u \text{ is unknown.} \]

- Split the data into 2 disjoint sets: training (\( \text{tr} \)) and test (\( \text{te} \)) sets.

**Procedure:**

1. Estimate \( \lambda_n \) with \( \hat{\lambda}_n^{(\text{tr})} \) (i.e., computed on the training set).
2. Optimize all \( \{(v_i, w_i)\}_{i=1}^J \) and Gaussian widths with gradient ascent.
3. Independence test with \( \hat{\lambda}_n^{(\text{te})} \). Reject \( H_0 \) if \( \hat{\lambda}_n^{(\text{te})} \geq T_\alpha \).

- Splitting avoids overfitting.
Optimization Procedure

- NFSIC\(^2 (X, Y) := \lambda_n := n u^\top \Sigma^{-1} u\) is unknown.
- Split the data into 2 disjoint sets: training (\(\text{tr}\)) and test (\(\text{te}\)) sets.

Procedure:

1. Estimate \(\lambda_n\) with \(\hat{\lambda}_n^{(\text{tr})}\) (i.e., computed on the training set).
2. Optimize all \(\{(v_i, w_i)\}_{i=1}^J\) and Gaussian widths with gradient ascent.
3. Independence test with \(\hat{\lambda}_n^{(\text{te})}\). Reject \(H_0\) if \(\hat{\lambda}_n^{(\text{te})} \geq T_\alpha\).

- Splitting avoids overfitting.

But, what does this do to \(\mathbb{P}(\hat{\lambda}_n \geq T_\alpha)\) when \(H_0\) holds?
Optimization Procedure

- **NFSIC**$^2(X, Y) := \lambda_n := n u^\top \Sigma^{-1} u$ is unknown.
- Split the data into 2 disjoint sets: training (tr) and test (te) sets.

Procedure:

1. Estimate $\lambda_n$ with $\hat{\lambda}_n^{(tr)}$ (i.e., computed on the training set).
2. Optimize all $\{(v_i, w_i)\}_{i=1}^J$ and Gaussian widths with gradient ascent.
3. Independence test with $\hat{\lambda}_n^{(te)}$. Reject $H_0$ if $\hat{\lambda}_n^{(te)} \geq T_\alpha$.

- Splitting avoids overfitting.

But, what does this do to $\mathbb{P}(\hat{\lambda}_n \geq T_\alpha)$ when $H_0$ holds?

- Still asymptotically at $\alpha$.
- $\lambda_n = 0$ iff $X, Y$ independent.
- So, under $H_0$, we do $\text{arg max } 0 = \text{arbitrary locations}$.
- Asymptotic null distribution is $\chi^2(J)$ for any locations.
Demo: 2D Rotation

\[ \hat{\mu}_{xy}(v, w) \]
Demo: 2D Rotation

\[ \hat{\mu}_{xy}(v, w) \]

\[ \hat{\mu}_x(v) \hat{\mu}_y(w) \]

\[ \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v) \hat{\mu}_y(w) \]

\[ \hat{\Sigma}(v, w) \]

\[ \hat{\lambda}_n \]
Demo: Sin Problem ($\omega = 1$)

$p(x, y) =$

\[ \hat{\mu}_{xy}(v, w) \]
Demo: Sin Problem ($\omega = 1$)

\[
p(x, y) =
\]
Simulation Settings

- \( n = \) full sample size
- All methods use Gaussian kernels for both \( X \) and \( Y \).

Compare 6 methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Tuning</th>
<th>Test size</th>
<th>Complex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSIC-opt</td>
<td>Proposed</td>
<td>Gradient descent</td>
<td>( n/2 )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>NFSIC-med</td>
<td>No tuning.</td>
<td>Random locations</td>
<td>( n )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>QHSIC</td>
<td>Full HSIC</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( O(n^2) )</td>
</tr>
<tr>
<td>NyHSIC</td>
<td>NyStrom HSIC</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>FHSIC</td>
<td>HSIC + RFFs*</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>RDC</td>
<td>RFFs + CCA</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( O(n \log n) )</td>
</tr>
</tbody>
</table>

*: Random Fourier features

- Given a problem, report rejection rate of \( H_0 \).
- 10 features for all (except QHSIC). \( J = 10 \) in NFSIC.
### Simulation Settings

- $n$ = full sample size
- All methods use Gaussian kernels for both $X$ and $Y$.

#### Compare 6 methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Tuning</th>
<th>Test size</th>
<th>Complex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSIC-opt</td>
<td>Proposed</td>
<td>Gradient descent</td>
<td>$n/2$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>NFSIC-med</td>
<td>No tuning.</td>
<td>Random locations</td>
<td>$n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>QHSIC</td>
<td>Full HSIC</td>
<td>Median heu.</td>
<td>$n$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>NyHSIC</td>
<td>NyStrom HSIC</td>
<td>Median heu.</td>
<td>$n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>FHSIC</td>
<td>HSIC + RFFs*</td>
<td>Median heu.</td>
<td>$n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>RDC</td>
<td>RFFs + CCA</td>
<td>Median heu.</td>
<td>$n$</td>
<td>$O(n \log n)$</td>
</tr>
</tbody>
</table>

* : Random Fourier features

- Given a problem, report rejection rate of $H_0$.
- 10 features for all (except QHSIC). $J = 10$ in NFSIC.
Simulation Settings

- \( n = \) full sample size
- All methods use Gaussian kernels for both \( X \) and \( Y \).

Compare 6 methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Tuning</th>
<th>Test size</th>
<th>Complex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSIC-opt</td>
<td>Proposed</td>
<td>Gradient descent</td>
<td>( n/2 )</td>
<td>( \mathcal{O}(n) )</td>
</tr>
<tr>
<td>NFSIC-med</td>
<td>No tuning.</td>
<td>Random locations</td>
<td>( n )</td>
<td>( \mathcal{O}(n) )</td>
</tr>
<tr>
<td>QHSIC</td>
<td>Full HSIC</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( \mathcal{O}(n^2) )</td>
</tr>
<tr>
<td>NyHSIC</td>
<td>NyStrom HSIC</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( \mathcal{O}(n) )</td>
</tr>
<tr>
<td>FHSIC</td>
<td>HSIC + RFFs*</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( \mathcal{O}(n) )</td>
</tr>
<tr>
<td>RDC</td>
<td>RFFs + CCA</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( \mathcal{O}(n \log n) )</td>
</tr>
</tbody>
</table>

\* : Random Fourier features

- Given a problem, report rejection rate of \( H_0 \).
- 10 features for all (except QHSIC). \( J = 10 \) in NFSIC.
Simulation Settings

- \( n = \) full sample size
- All methods use Gaussian kernels for both \( X \) and \( Y \).

Compare 6 methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Tuning</th>
<th>Test size</th>
<th>Complex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSIC-opt</td>
<td>Proposed</td>
<td>Gradient descent</td>
<td>( n/2 )</td>
<td>( \mathcal{O}(n) )</td>
</tr>
<tr>
<td>NFSIC-med</td>
<td>No tuning.</td>
<td>Random locations</td>
<td>( n )</td>
<td>( \mathcal{O}(n) )</td>
</tr>
<tr>
<td>QHSIC</td>
<td>Full HSIC</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( \mathcal{O}(n^2) )</td>
</tr>
<tr>
<td>NyHSIC</td>
<td>NyStrom HSIC</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( \mathcal{O}(n) )</td>
</tr>
<tr>
<td>FHSIC</td>
<td>HSIC + RFFs*</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( \mathcal{O}(n) )</td>
</tr>
<tr>
<td>RDC</td>
<td>RFFs + CCA</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( \mathcal{O}(n \log n) )</td>
</tr>
</tbody>
</table>

* : Random Fourier features

- Given a problem, report rejection rate of \( H_0 \).
- 10 features for all (except QHSIC). \( J = 10 \) in NFSIC.
Simulation Settings

- \( n = \) full sample size
- All methods use Gaussian kernels for both \( X \) and \( Y \).

Compare 6 methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Tuning</th>
<th>Test size</th>
<th>Complex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSIC-opt</td>
<td>Proposed</td>
<td>Gradient descent</td>
<td>( n/2 )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>NFSIC-med</td>
<td>No tuning.</td>
<td>Random locations</td>
<td>( n )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>QHSIC</td>
<td>Full HSIC</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( O(n^2) )</td>
</tr>
<tr>
<td>NyHSIC</td>
<td>NyStrom HSIC</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>FHSIC</td>
<td>HSIC + RFFs*</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>RDC</td>
<td>RFFs + CCA</td>
<td>Median heu.</td>
<td>( n )</td>
<td>( O(n \log n) )</td>
</tr>
</tbody>
</table>

*: Random Fourier features

- Given a problem, report rejection rate of \( H_0 \).
- 10 features for all (except QHSIC). \( J = 10 \) in NFSIC.
Simulation Settings

- $n = \text{full sample size}$
- All methods use Gaussian kernels for both $X$ and $Y$.

Compare 6 methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Tuning</th>
<th>Test size</th>
<th>Complex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSIC-opt</td>
<td>Proposed</td>
<td>Gradient descent</td>
<td>$n/2$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>NFSIC-med</td>
<td>No tuning.</td>
<td>Random locations</td>
<td>$n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>QHSIC</td>
<td>Full HSIC</td>
<td>Median heu.</td>
<td>$n$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>NyHSIC</td>
<td>NyStrom HSIC</td>
<td>Median heu.</td>
<td>$n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>FHSIC</td>
<td>HSIC + RFFs*</td>
<td>Median heu.</td>
<td>$n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>RDC</td>
<td>RFFs + CCA</td>
<td>Median heu.</td>
<td>$n$</td>
<td>$O(n \log n)$</td>
</tr>
</tbody>
</table>

*: Random Fourier features

- Given a problem, report rejection rate of $H_0$.
- 10 features for all (except QHSIC). $J = 10$ in NFSIC.
Toy Problem 1: Independent Gaussians

- \( X \sim \mathcal{N}(0, I_{d_x}) \) and \( Y \sim \mathcal{N}(0, I_{d_y}) \).
- Independent \( X, Y \). So, \( H_0 \) holds.
- Set \( \alpha := 0.05, d_x = d_y = 250 \).
Toy Problem 1: Independent Gaussians

- \( X \sim \mathcal{N}(0, I_{d_x}) \) and \( Y \sim \mathcal{N}(0, I_{d_y}) \).
- Independent \( X, Y \). So, \( H_0 \) holds.
- Set \( \alpha := 0.05, d_x = d_y = 250 \).

Correct type-I errors (false positive rate).
Toy Problem 1: Independent Gaussians

- \( X \sim \mathcal{N}(0, I_{d_x}) \) and \( Y \sim \mathcal{N}(0, I_{d_y}) \).
- Independent \( X, Y \). So, \( H_0 \) holds.
- Set \( \alpha := 0.05, d_x = d_y = 250 \).

Correct type-I errors (false positive rate).
Toy Problem 2: Sinusoid

- \( p_{xy}(x, y) \propto 1 + \sin(\omega x) \sin(\omega y) \) where \( x, y \in (-\pi, \pi) \).
- Local changes between \( p_{xy} \) and \( p_x p_y \).
- Set \( n = 4000 \).

Main Point: NFSIC can handle well the local changes in the joint space.
Toy Problem 2: Sinusoid

- $p_{xy}(x, y) \propto 1 + \sin(\omega x) \sin(\omega y)$ where $x, y \in (-\pi, \pi)$.
- Local changes between $p_{xy}$ and $p_x p_y$.
- Set $n = 4000$. 

![Contour plot of $\omega = 1.00$](image)
**Toy Problem 2: Sinusoid**

- \( p_{xy}(x, y) \propto 1 + \sin(\omega x) \sin(\omega y) \) where \( x, y \in (-\pi, \pi) \).
- Local changes between \( p_{xy} \) and \( p_x p_y \).
- Set \( n = 4000 \).
**Toy Problem 2: Sinusoid**

- \( p_{xy}(x, y) \propto 1 + \sin(\omega x) \sin(\omega y) \) where \( x, y \in (-\pi, \pi) \).

- Local changes between \( p_{xy} \) and \( p_x p_y \).

- Set \( n = 4000 \).

![Graph showing the sinusoidal distribution with \( \omega = 3.00 \)]
Toy Problem 2: Sinusoid

- $p_{xy}(x, y) \propto 1 + \sin(\omega x) \sin(\omega y)$ where $x, y \in (-\pi, \pi)$.
- Local changes between $p_{xy}$ and $p_x p_y$.
- Set $n = 4000$. 

![Diagram](image-url)
Toy Problem 2: Sinusoid

- \( p_{xy}(x, y) \propto 1 + \sin(\omega x) \sin(\omega y) \) where \( x, y \in (-\pi, \pi) \).
- Local changes between \( p_{xy} \) and \( p_x p_y \).
- Set \( n = 4000 \).
Toy Problem 2: Sinusoid

- $p_{xy}(x, y) \propto 1 + \sin(\omega x)\sin(\omega y)$ where $x, y \in (-\pi, \pi)$.
- Local changes between $p_{xy}$ and $p_x p_y$.
- Set $n = 4000$.

Main Point: NFSIC can handle well the local changes in the joint space.
**Toy Problem 2: Sinusoid**

- $p_{xy}(x, y) \propto 1 + \sin(\omega x) \sin(\omega y)$ where $x, y \in (-\pi, \pi)$.
- Local changes between $p_{xy}$ and $p_x p_y$.
- Set $n = 4000$.

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>Test power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Main Point: NFSIC can handle well the local changes in the joint space.
Toy Problem 3: Gaussian Sign

- $y = |Z| \prod_{i=1}^{d_x} \text{sign}(x_i)$, where $x \sim \mathcal{N}(0, I_{d_y})$ and $Z \sim \mathcal{N}(0, 1)$ (noise).
- Full interaction among $x_1, \ldots, x_{d_x}$.
- Need to consider all $x_1, \ldots, x_d$ to detect the dependency.

Main Point: NFSIC can handle feature interaction.
Toy Problem 3: Gaussian Sign

- $y = |Z| \prod_{i=1}^{d_x} \text{sign}(x_i)$, where $x \sim \mathcal{N}(0, I_{d_y})$ and $Z \sim \mathcal{N}(0, 1)$ (noise).
- Full interaction among $x_1, \ldots, x_{d_x}$.
- Need to consider all $x_1, \ldots, x_d$ to detect the dependency.

Main Point: NFSIC can handle feature interaction.
HSIC vs. FSIC

Recall the witness

\[ \hat{u}(v, w) = \hat{\mu}_{xy}(v, w) - \hat{\mu}_x(v)\hat{\mu}_y(w). \]

**HSIC** [Gretton et al., 2005]

\[ = \|\hat{u}\|_{\text{RKHS}} \]

Good when difference between \( p_{xy} \) and \( p_x p_y \) is spatially diffuse.

- \( \hat{u} \) is almost flat.

**FSIC** [proposed]

\[ = \frac{1}{J} \sum_{i=1}^{J} \hat{u}^2(v_i, w_i) \]

Good when difference between \( p_{xy} \) and \( p_x p_y \) is local.

- \( \hat{u} \) is mostly zero, has many peaks (feature interaction).
Real Problem 1: Million Song Data

Song \((X)\) vs. year of release \((Y)\).

- Western commercial tracks from 1922 to 2011 [Bertin-Mahieux et al., 2011].
- \(X \in \mathbb{R}^{90}\) contains audio features.
- \(Y \in \mathbb{R}\) is the year of release.
Real Problem 1: Million Song Data

Song \((X)\) vs. year of release \((Y)\).

- Western commercial tracks from 1922 to 2011
  - [Bertin-Mahieux et al., 2011].
- \(X \in \mathbb{R}^{90}\) contains audio features.
- \(Y \in \mathbb{R}\) is the year of release.

<table>
<thead>
<tr>
<th>Sample size (n)</th>
<th>Type-I error</th>
<th>Test power</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.000</td>
<td>0.3</td>
</tr>
<tr>
<td>1000</td>
<td>0.005</td>
<td>0.6</td>
</tr>
<tr>
<td>1500</td>
<td>0.010</td>
<td>0.8</td>
</tr>
<tr>
<td>2000</td>
<td>0.015</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Break \((X, Y)\) pairs to simulate \(H_0\).

- \(H_1\) is true.
Real Problem 2: Videos and Captions

Youtube video ($X$) vs. caption ($Y$).

- VideoStory46K [Habibian et al., 2014]
- $Y \in \mathbb{R}^{1878}$: bag of words. TF.
Real Problem 2: Videos and Captions

Youtube video \((X)\) vs. caption \((Y)\).

- VideoStory46K [Habibian et al., 2014]
- \(X \in \mathbb{R}^{2000}\): Fisher vector encoding of motion boundary histograms descriptors [Wang and Schmid, 2013].
- \(Y \in \mathbb{R}^{1878}\): bag of words. TF.

<table>
<thead>
<tr>
<th>Sample size</th>
<th>Type-I error</th>
<th>Test power</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>0.002</td>
<td>0.0</td>
</tr>
<tr>
<td>4000</td>
<td>0.004</td>
<td>0.2</td>
</tr>
<tr>
<td>6000</td>
<td>0.006</td>
<td>0.4</td>
</tr>
<tr>
<td>8000</td>
<td>0.008</td>
<td>0.6</td>
</tr>
</tbody>
</table>

- Break \((X, Y)\) pairs to simulate \(H_0\).
- \(H_1\) is true.
Penalize Redundant Test Locations

- Consider the Sin problem. Use $J = 2$ locations.
- Optimization objective: $\hat{\lambda}_n$.
- Write $t = (v, w)$. Fix $t_1$ at ★. Plot $t_2 \rightarrow \hat{\lambda}_n(t_1, t_2)$.

The optimized $t_1, t_2$ will not be in the same neighbourhood.
Test Power vs. $J$

- Test power *does not* always increase with $J$ (number of test locations).
- $n = 800$.

Accurate estimation of $\hat{\Sigma} \in \mathbb{R}^{J \times J}$ in $\hat{\lambda}_n = n \hat{u}^T \left( \hat{\Sigma} + \gamma_n I \right)^{-1} \hat{u}$ becomes more difficult.

- Large $J$ defeats the purpose of a linear-time test.
Conclusions

- Proposed The Finite Set Independence Criterion (FSIC).
- Independence test based on FSIC is
  1. non-parametric,
  2. linear-time,
  3. adaptive (parameters are automatically tuned).

Future works

- Any way to interpret the learned \( \{(v_i, w_i)\}_{i=1}^J \)?
- Relative efficiency of FSIC vs. block HSIC, RFF-HSIC.

https://github.com/wittawatj/fsic-test
Conclusions

- Proposed The Finite Set Independence Criterion (FSIC).
- Independence test based on FSIC is
  1. non-parametric,
  2. linear-time,
  3. adaptive (parameters automatically tuned).

Future works

- Any way to interpret the learned \( \{(v_i, w_i)\}_{i=1}^J \)?
- Relative efficiency of FSIC vs. block HSIC, RFF-HSIC.

https://github.com/wittawatj/fsic-test
Questions?

Thank you


