Interpretable Distribution Features with Maximum Testing Power

Wittawat Jitkrittum,
Zoltán Szabó,
Kacper Chwialkowski,
Arthur Gretton

Gatsby Computational Neuroscience Unit,
University College London

NIPS 2016, Barcelona Spain

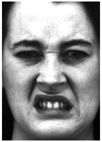
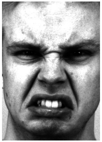
Overview

- **Have:** Two collections of samples X, Y from unknown distributions P and Q.

Positive emotions

$\mathsf{X} = \{$ , , , $\ldots\} \sim P$

Negative emotions

$\mathsf{Y} = \{$ , , , $\ldots\} \sim Q$

- **Goal:** Learn distinguishing features that indicate how P and Q differ.

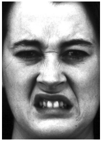
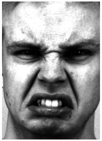
Overview

- **Have:** Two collections of samples X, Y from unknown distributions P and Q.

Positive emotions

$\mathsf{X} = \{$ , ,  $, \ldots \} \sim P$

Negative emotions

$\mathsf{Y} = \{$ , ,  $, \ldots \} \sim Q$

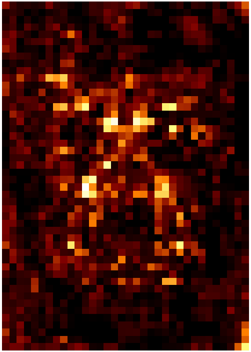
- **Goal:** Learn distinguishing features that indicate how P and Q differ.

Overview

From the two collections

{ , , , . . .} and { , , , . . .},

produce a new point indicating where to look for the differences



Overview

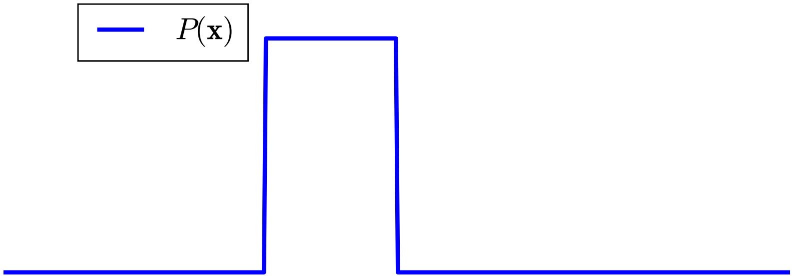
From the two collections

{ , , , ... } and { , , , ... },

produce a new point indicating where to look for the differences

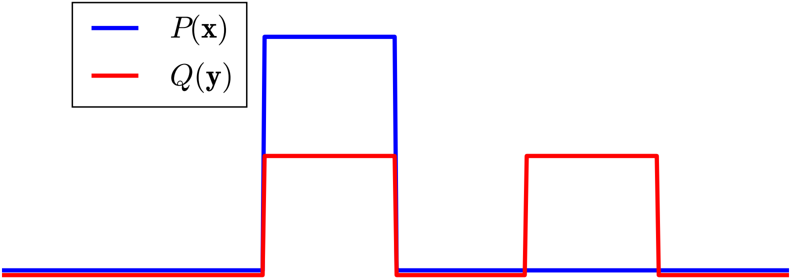
Distinguishing Feature(s)

Where is the best location to observe the difference of $P(\mathbf{x})$ and $Q(\mathbf{y})$?



Distinguishing Feature(s)

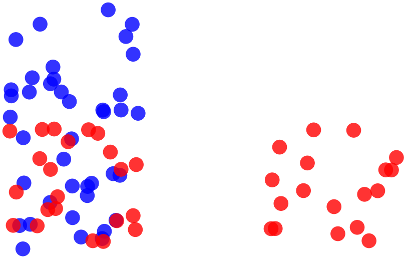
Where is the best location to observe the difference of $P(\mathbf{x})$ and $Q(\mathbf{y})$?



- **Why:** best location = distinguishing feature.
- **Propose:** a linear-time algorithm to find such data-driven feature(s).

Distinguishing Feature(s)

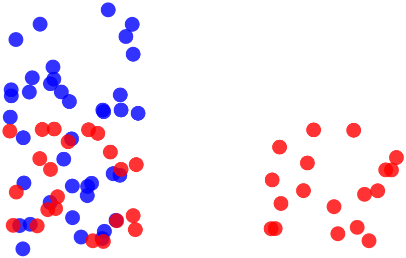
Where is the best location to observe the difference of $P(\mathbf{x})$ and $Q(\mathbf{y})$?



- **Why:** best location = distinguishing feature.
- **Propose:** a linear-time algorithm to find such data-driven feature(s).

Distinguishing Feature(s)

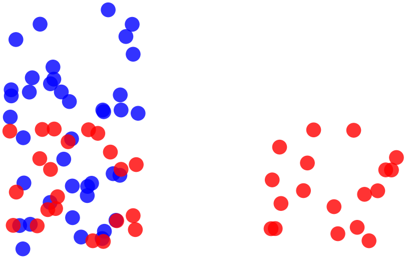
Where is the best location to observe the difference of $P(\mathbf{x})$ and $Q(\mathbf{y})$?



- **Why:** best location = distinguishing feature.
- **Propose:** a linear-time algorithm to find such data-driven feature(s).

Distinguishing Feature(s)

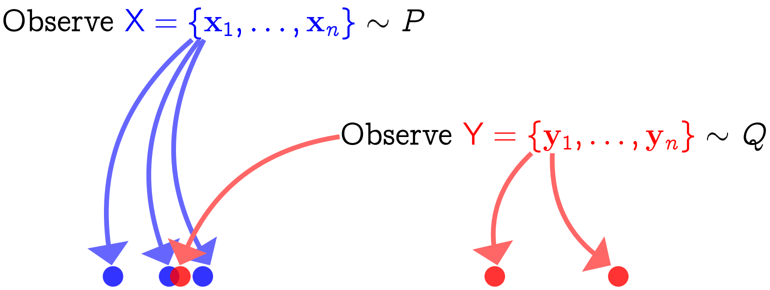
Where is the best location to observe the difference of $P(\mathbf{x})$ and $Q(\mathbf{y})$?



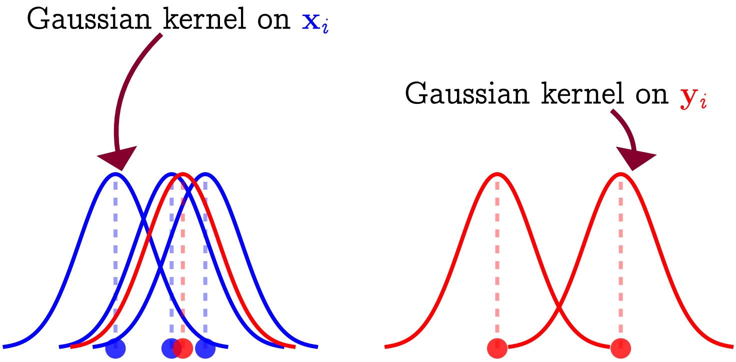
- **Why:** best location = distinguishing feature.
- **Propose:** a linear-time algorithm to find such data-driven feature(s).

Witness Function (Gretton et al., 2012)

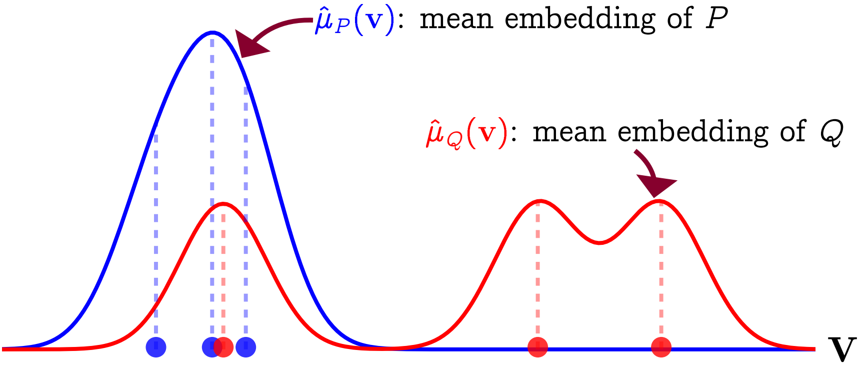
Witness Function (Gretton et al., 2012)



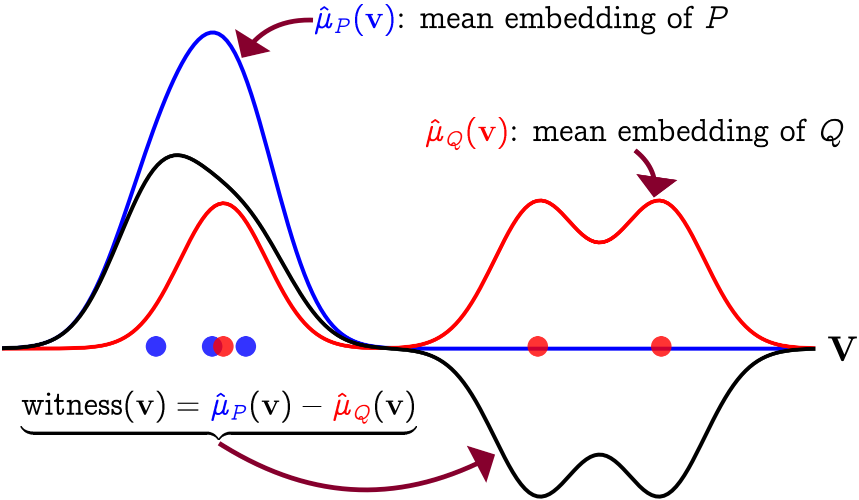
Witness Function (Gretton et al., 2012)



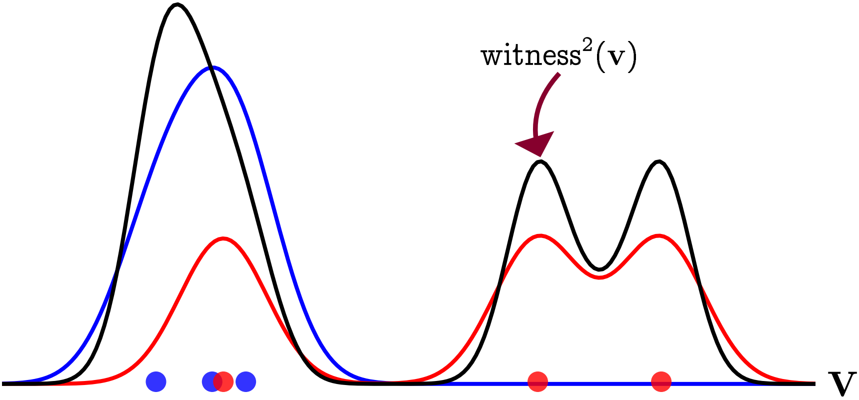
Witness Function (Gretton et al., 2012)



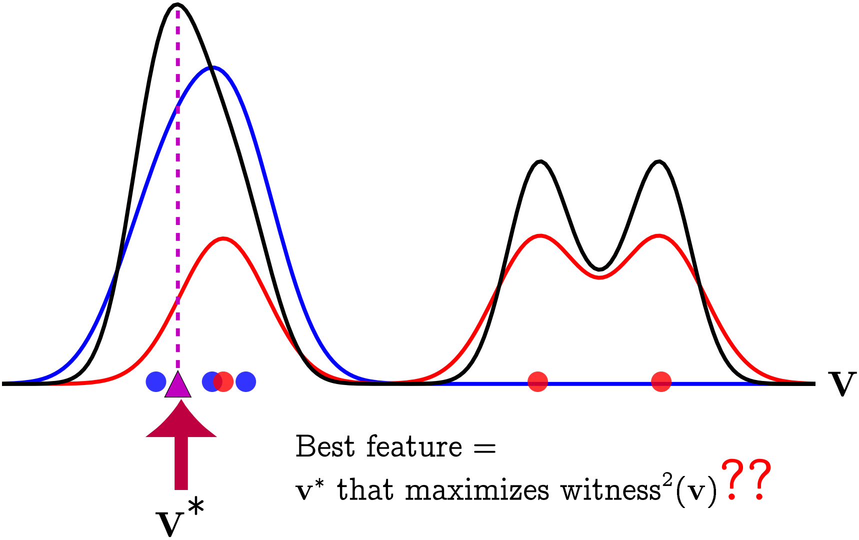
Witness Function (Gretton et al., 2012)



Witness Function (Gretton et al., 2012)

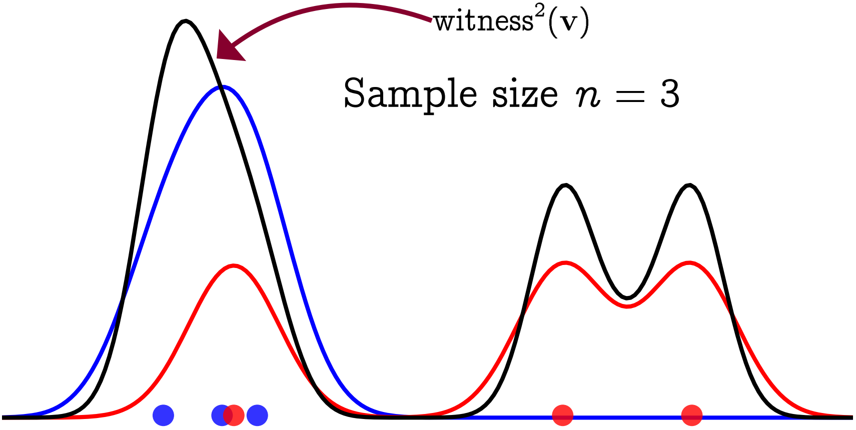


Witness Function (Gretton et al., 2012)

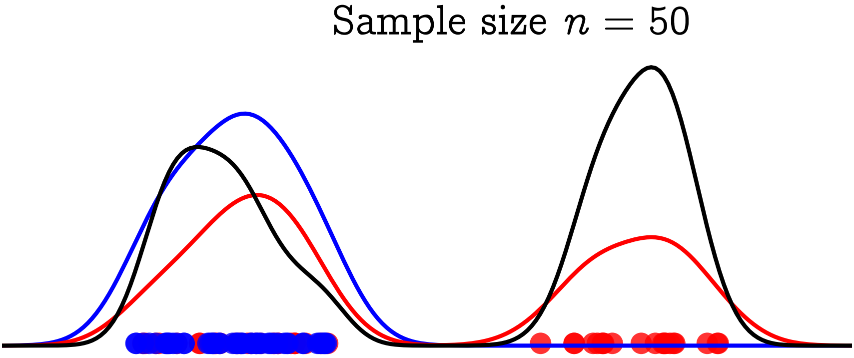


Best feature =
v* that maximizes witness^2(v)??

Failure Mode of the Witness Function

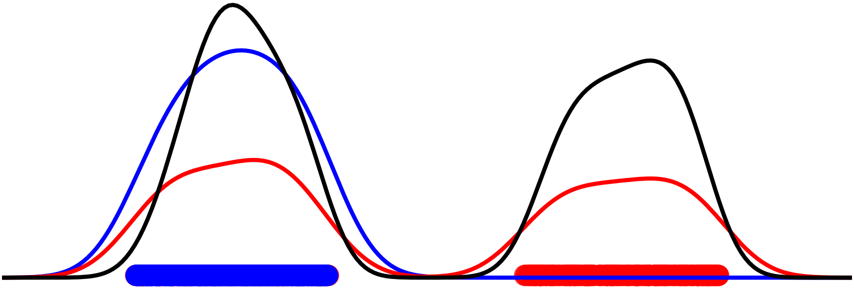


Failure Mode of the Witness Function



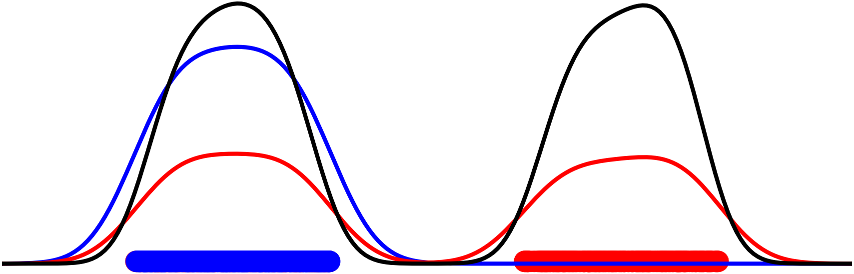
Failure Mode of the Witness Function

Sample size $n = 500$

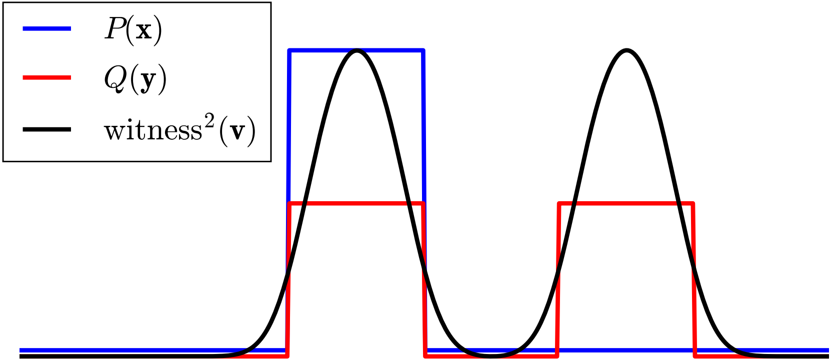


Failure Mode of the Witness Function

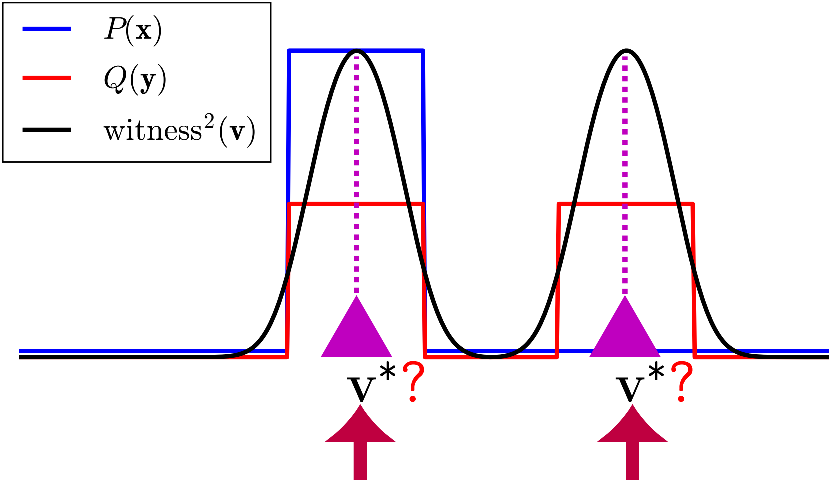
Sample size $n = 5000$



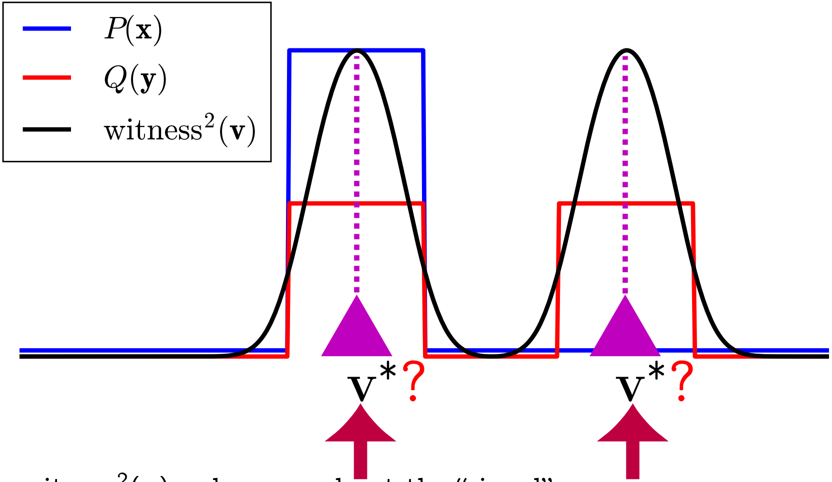
Failure Mode of the Witness Function



Failure Mode of the Witness Function



Failure Mode of the Witness Function



- $\text{witness}^2(\mathbf{v})$ only cares about the “signal”.
- Not the “noise” (variability) at each feature.

The ME (Mean Embeddings) Statistic (Chwialkowski et al., 2015)

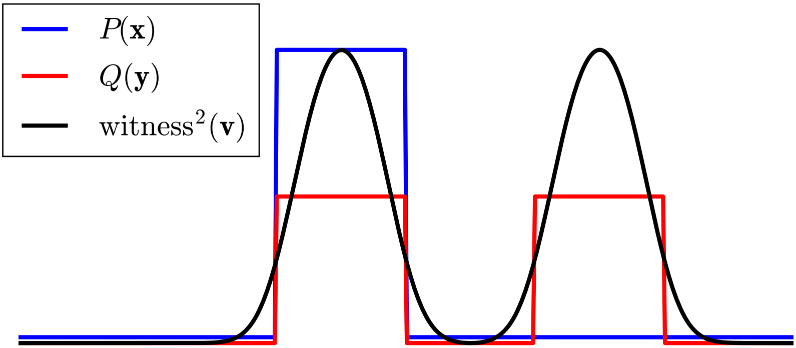
- **Variance of $\mathbf{v}$** = variance of $\mathbf{v}$ from X + variance of $\mathbf{v}$ from Y.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.

The ME (Mean Embeddings) Statistic (Chwialkowski et al., 2015)

- **Variance of $\mathbf{v}$** = variance of $\mathbf{v}$ from X + variance of $\mathbf{v}$ from Y.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.

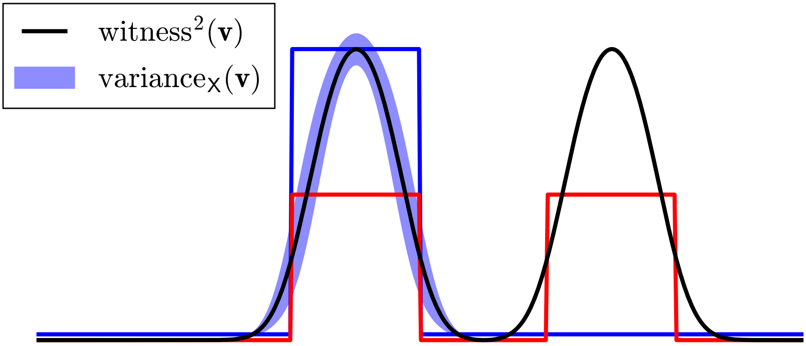
The ME (Mean Embeddings) Statistic (Chwialkowski et al., 2015)

- Variance of $\mathbf{v}$ = variance of $\mathbf{v}$ from X + variance of $\mathbf{v}$ from Y.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.



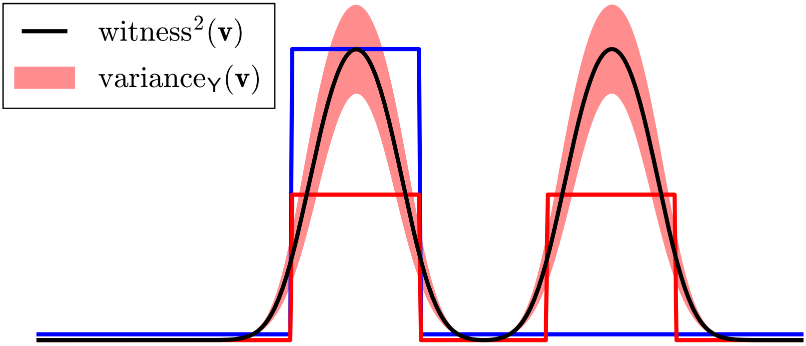
The ME (Mean Embeddings) Statistic (Chwialkowski et al., 2015)

- Variance of $\mathbf{v}$ = variance of $\mathbf{v}$ from X + variance of $\mathbf{v}$ from Y.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.



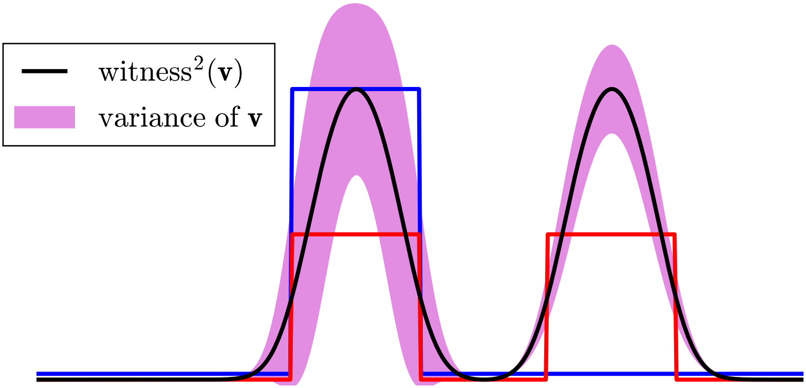
The ME (Mean Embeddings) Statistic (Chwialkowski et al., 2015)

- Variance of $\mathbf{v}$ = variance of $\mathbf{v}$ from X + variance of $\mathbf{v}$ from Y.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.



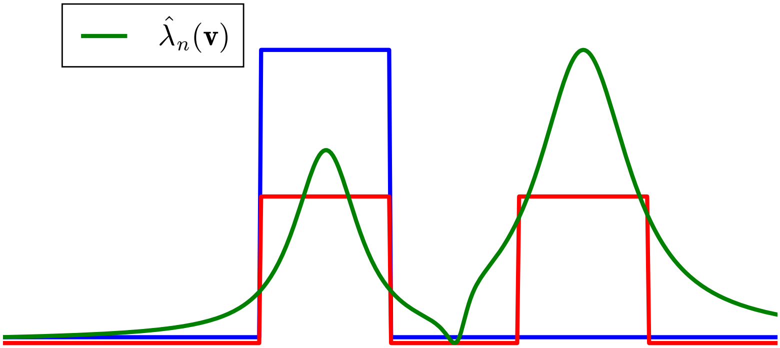
The ME (Mean Embeddings) Statistic (Chwialkowski et al., 2015)

- Variance of $\mathbf{v}$ = variance of $\mathbf{v}$ from X + variance of $\mathbf{v}$ from Y.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.



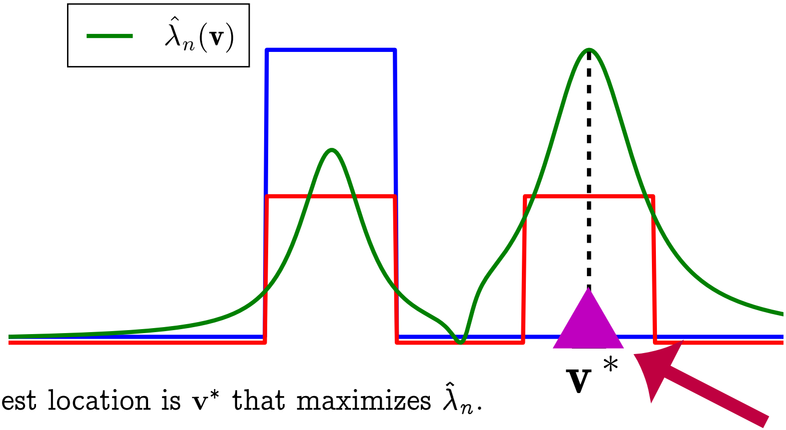
The ME (Mean Embeddings) Statistic (Chwialkowski et al., 2015)

- Variance of $\mathbf{v}$ = variance of $\mathbf{v}$ from X + variance of $\mathbf{v}$ from Y.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.



The ME (Mean Embeddings) Statistic (Chwialkowski et al., 2015)

- Variance of $\mathbf{v}$ = variance of $\mathbf{v}$ from X + variance of $\mathbf{v}$ from Y.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.



- Best location is $\mathbf{v}^*$ that maximizes $\hat{\lambda}_n$.

Properties of the ME Statistic

- Can construct a two-sample test using J features.
  - $H_0 : P = Q$ vs. $H_1 : P \neq Q$.
- Choosing the best J features increases a lower bound on the test power.
  - Test power $= \mathbb{P}(\text{reject } H_0 \mid H_1 \text{ is true})$.
- **Runtime:** $\mathcal{O}(n)$. Fast.

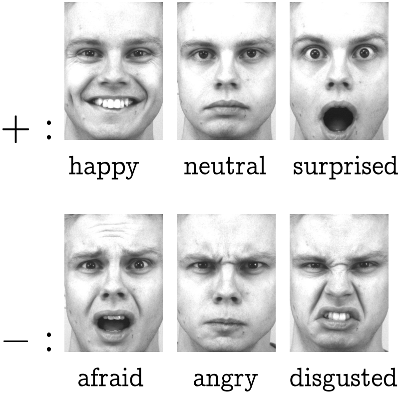
Properties of the ME Statistic

- Can construct a two-sample test using J features.
  - $H_0 : P = Q$ vs. $H_1 : P \neq Q$.
- Choosing the best J features increases a lower bound on the test power.
  - Test power $= \mathbb{P}(\text{reject } H_0 \mid H_1 \text{ is true})$.
- Runtime: $\mathcal{O}(n)$. Fast.

Properties of the ME Statistic

- Can construct a two-sample test using J features.
  - $H_0 : P = Q$ vs. $H_1 : P \neq Q$.
- Choosing the best J features increases a lower bound on the test power.
  - Test power $= \mathbb{P}(\text{reject } H_0 \mid H_1 \text{ is true})$.
- **Runtime**: $\mathcal{O}(n)$. Fast.

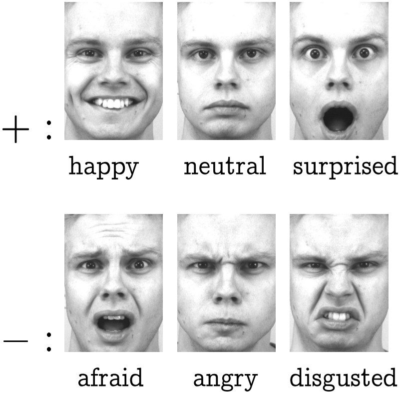
Distinguishing Positive/Negative Emotions

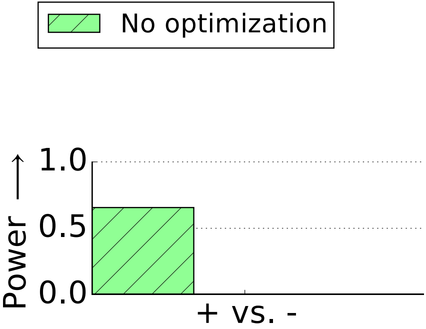


- 35 females and 35 males (Lundqvist et al., 1998).
- $48 \times 34 = 1632$ dimensions. Pixel features.
- $n = 201$.

- Test power comparable to the state-of-the-art MMD test.
- Informative features: differences at the nose, and smile lines.

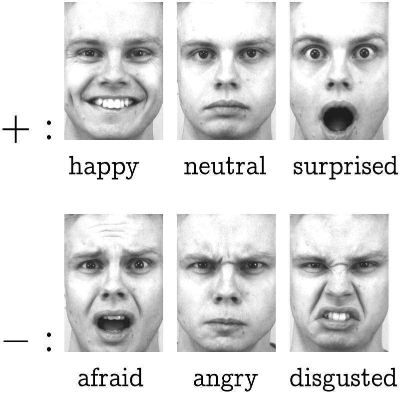
Distinguishing Positive/Negative Emotions

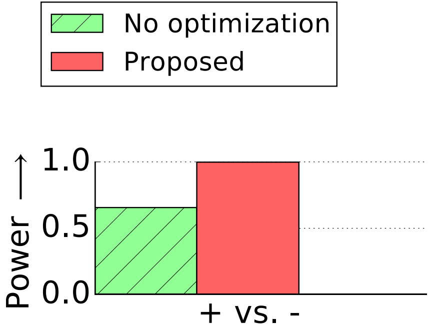




- Test power comparable to the state-of-the-art MMD test.
- Informative features: differences at the nose, and smile lines.

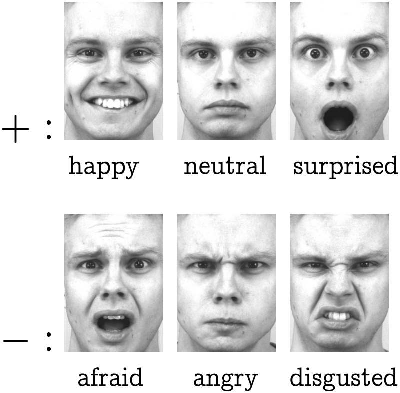
Distinguishing Positive/Negative Emotions

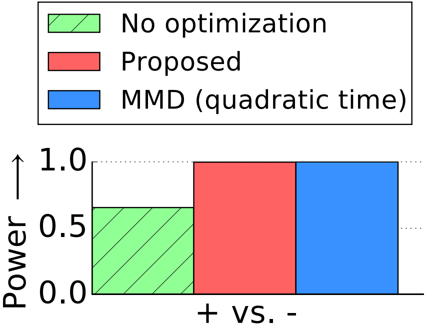




- Test power comparable to the state-of-the-art MMD test.
- Informative features: differences at the nose, and smile lines.

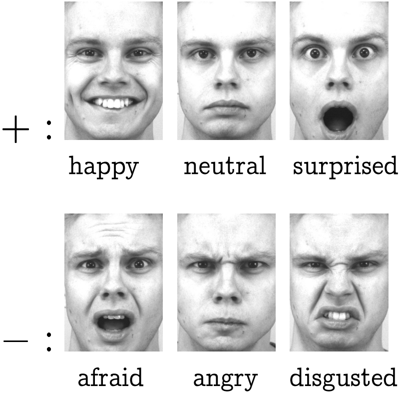
Distinguishing Positive/Negative Emotions

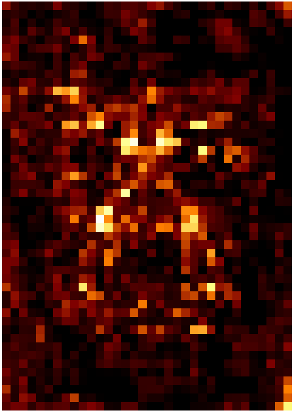




- Test power comparable to the state-of-the-art MMD test.
- Informative features: differences at the nose, and smile lines.

Distinguishing Positive/Negative Emotions

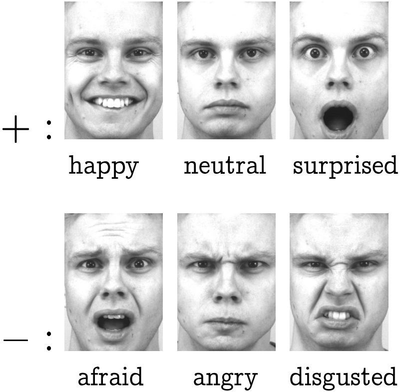




Learned feature

- Test power comparable to the state-of-the-art MMD test.
- **Informative features**: differences at the nose, and smile lines.

Distinguishing Positive/Negative Emotions



Learned feature

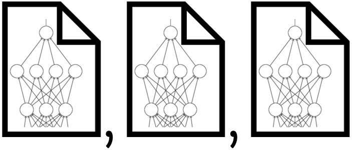
- Test power comparable to the state-of-the-art MMD test.
- **Informative features**: differences at the nose, and smile lines.

Bayesian Inference Vs. Deep Learning Papers

Papers on Bayesian inference

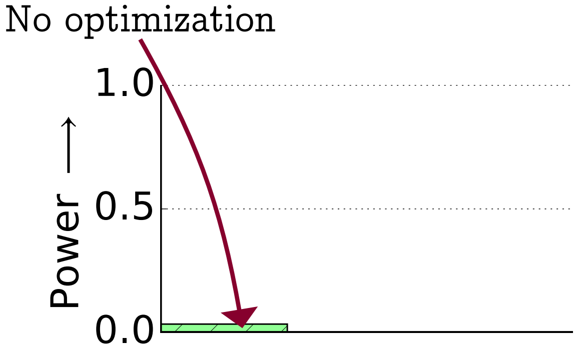
$\mathsf{X} = \{$  $, \ldots\} \sim P$

Papers on deep learning

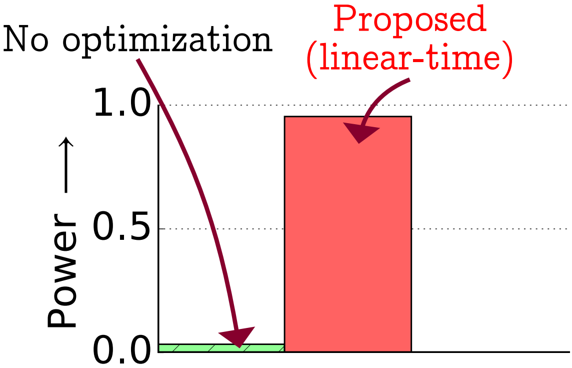
$\mathsf{Y} = \{$  $, \ldots\} \sim Q$

- NIPS papers (1988-2015)
- Sample size $n = 216$.
- Random 2000 nouns (dimensions). TF-IDF representation.

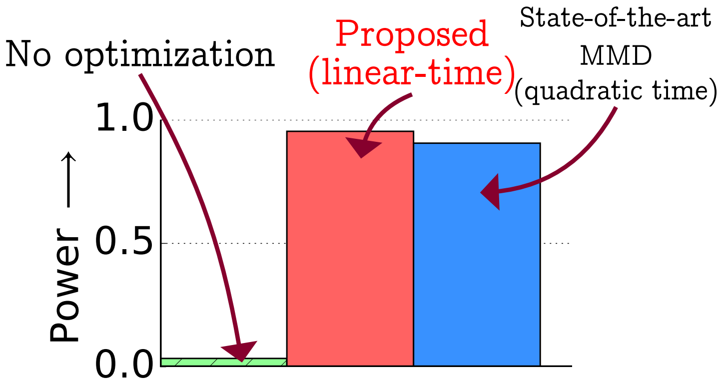
Bayesian Inference Vs. Deep Learning Papers



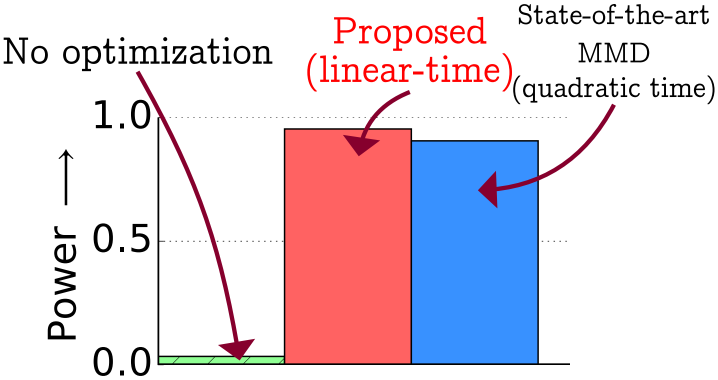
Bayesian Inference Vs. Deep Learning Papers



Bayesian Inference Vs. Deep Learning Papers



Bayesian Inference Vs. Deep Learning Papers



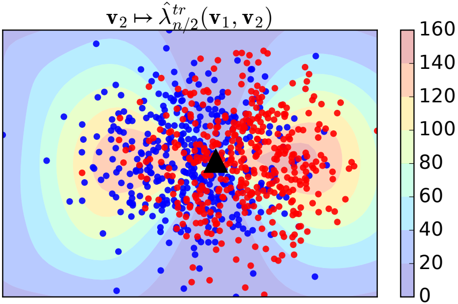
Learned informative feature (a new document):

infer, Bayes, Monte Carlo, adaptor, motif,
haplotype, ECG, covariance, Boltzmann

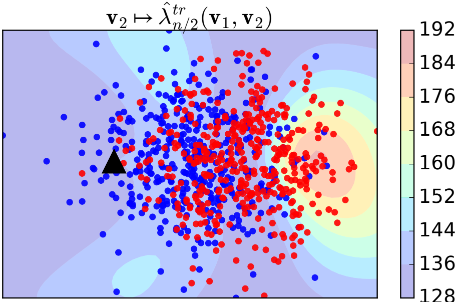
Illustration: Two Informative Features

- 2D problem.

$$P : \mathcal{N}([0,0], I)$$
$$Q : \mathcal{N}([1,0], I)$$



- $J = 2$ features.
- Fix $\mathbf{v}_1$ to ▲.
- Contour plot of $\mathbf{v}_2 \mapsto \hat{\lambda}_n(\{\mathbf{v}_1, \mathbf{v}_2\})$.
- $\{\mathbf{v}_1, \mathbf{v}_2\}$ chosen to reveal the difference of P and Q.



Summary

$+$:

$-$:

$\Longrightarrow$

Learned feature

Fast method to extract features
for distinguishing two distributions

- Python code available: http://wittawat.com

Questions?

Thank you

Full ME Test Statistic

- Let $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_J\}$ be the J test locations.
- Let $\overline{\mathbf{z}}_n := \begin{pmatrix} \hat{\mu}_P(\mathbf{v}_1) - \hat{\mu}_Q(\mathbf{v}_1) \\ \vdots \\ \hat{\mu}_P(\mathbf{v}_J) - \hat{\mu}_Q(\mathbf{v}_J) \end{pmatrix} \in \mathbb{R}^J$.
- Let $(\mathbf{S}_n)_{ij} := \widehat{\mathrm{cov}}_{\mathbf{x}}[k(\mathbf{x}, \mathbf{v}_i), k(\mathbf{x}, \mathbf{v}_j)] + \widehat{\mathrm{cov}}_{\mathbf{y}}[k(\mathbf{y}, \mathbf{v}_i), k(\mathbf{y}, \mathbf{v}_j)] \in \mathbb{R}^{J \times J}$.
- Then, the statistic

$$\hat{\lambda}_n := n\overline{\mathbf{z}}_n^\top (\mathbf{S}_n + \gamma_n I)^{-1} \overline{\mathbf{z}}_n,$$

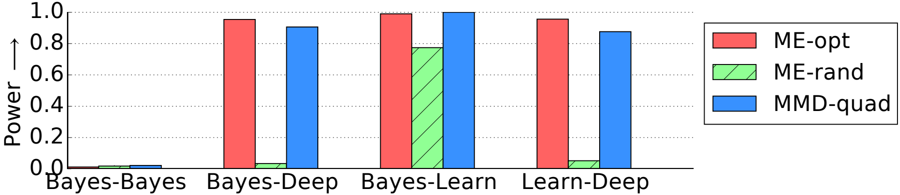
  where $\gamma_n > 0$ is a regularization parameter.
- When $J = 1$,

$$\hat{\lambda}_n = n \frac{[\hat{\mu}_P(\mathbf{v}) - \hat{\mu}_Q(\mathbf{v})]^2}{\gamma_\mathrm{n} + \mathrm{var}_{\mathbf{x}}[k(\mathbf{x}, \mathbf{v})] + \mathrm{var}_{\mathbf{y}}[k(\mathbf{y}, \mathbf{v})]}.$$

- Computing $\hat{\lambda}_n$: $\mathcal{O}(J^3 + J^2 n + Jdn)$.
- Optimization of $\mathcal{V}$: $\mathcal{O}(J^3 + J^2 dn)$.

Distinguishing NIPS Articles

- Bayesian inference, Deep learning, Learning theory
- Random 2000 nouns (dimensions). TF-IDF representation.

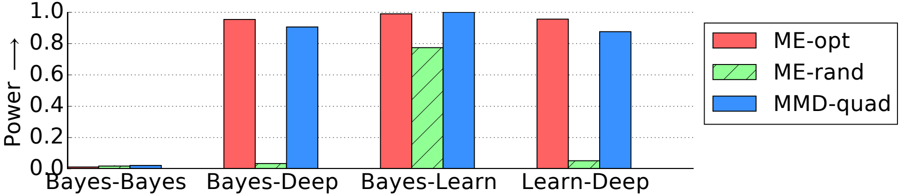


Learned informative features (bags of words):

Bayes-Deep: infer, Bayes, Monte Carlo, adaptor, motif, haplotype, ECG
Bayes-Learn: infer, Markov, graphic, segment, bandit, boundary, favor
Learn-Deep: deep, forward, delay, subgroup, bandit, receptor, invariance

Distinguishing NIPS Articles

- Bayesian inference, Deep learning, Learning theory
- Random 2000 nouns (dimensions). TF-IDF representation.



Learned informative features (bags of words):

Bayes-Deep: infer, Bayes, Monte Carlo, adaptor, motif, haplotype, ECG
Bayes-Learn: infer, Markov, graphic, segment, bandit, boundary, favor
Learn-Deep: deep, forward, delay, subgroup, bandit, receptor, invariance

Preprocessing of NIPS articles

- Remove stop words, and stem.
- A paper belongs to a group if it has at least one keyword.

1. **Bayesian inference** (Bayes): graphical model, bayesian, inference, mcmc, monte carlo, posterior, prior, variational, markov, latent, probabilistic, exponential family.
2. **Deep learning** (Deep): deep, drop out, auto-encod, convolutional, neural net, belief net, boltzmann.
3. **Learning theory** (Learn): learning theory, consistency, theoretical guarantee, complexity, pac-bayes, pac-learning, generalization, uniform converg, bound, deviation, inequality, risk min, minimax, structural risk, VC, rademacher, asymptotic.
4. **Neuroscience** (Neuro): motor control, neural, neuron, spiking, spike, cortex, plasticity, neural decod, neural encod, brain imag, biolog, perception, cognitive, emotion, synap, neural population, cortical, firing rate, firing-rate, sensor.

Lower Bound on Test Power

- Let $\mathcal{K}$ be a kernel class such that $\sup_{k\in\mathcal{K}}\sup_{(\mathbf{x},\mathbf{y})\in\mathcal{X}^2}|k(\mathbf{x},\mathbf{y})|\leq B$.
- Let $\mathbb{V}$ be a collection in which each element is a set of J test locations.
- Assume $\tilde{c} := \sup_{\mathcal{V}\in\mathbb{V},k\in\mathcal{K}}\|\Sigma^{-1}\|_F < \infty$.

Proposition

The test power $\mathbb{P}_{H_1}\left(\hat{\lambda}_n \geq T_\alpha\right)$ *of the ME test satisfies* $\mathbb{P}_{H_1}\left(\hat{\lambda}_n \geq T_\alpha\right) \geq L(\lambda_n)$ *where*

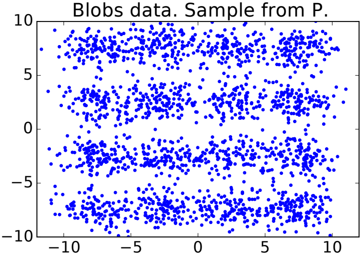
$$L(\lambda_n) := 1 - 2e^{-\xi_1(\lambda_n - T_\alpha)^2/n} - 2e^{-\frac{[\gamma_n(\lambda_n - T_\alpha)(n-1)-\xi_2 n]^2}{\xi_3 n(2n-1)^2}} - 2e^{-[(\lambda_n - T_\alpha)/3 - \bar{c}_3 n\gamma_n]^2\gamma_n^2/\xi_4},$$

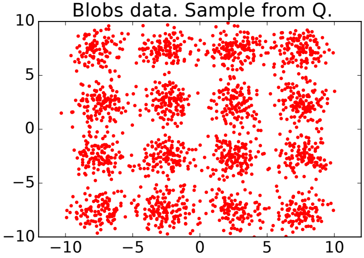
and $\bar{c}_3, \xi_1, \ldots \xi_4$ *are positive constants depending on only* B, J *and* $\tilde{c}$. *For large* n, $L(\lambda_n)$ *is increasing in* λ_n.

- $\lambda_n := n\mu^\top\Sigma^{-1}\mu$ is the population counterpart of $\hat{\lambda}_n$.
- $\mu = \mathbb{E}_{\mathbf{xy}}[\mathbf{z}_1]$ and $\Sigma = \mathbb{E}_{\mathbf{xy}}[(\mathbf{z}_1-\mu)(\mathbf{z}_1-\mu)^\top]$.

Four Toy Problems

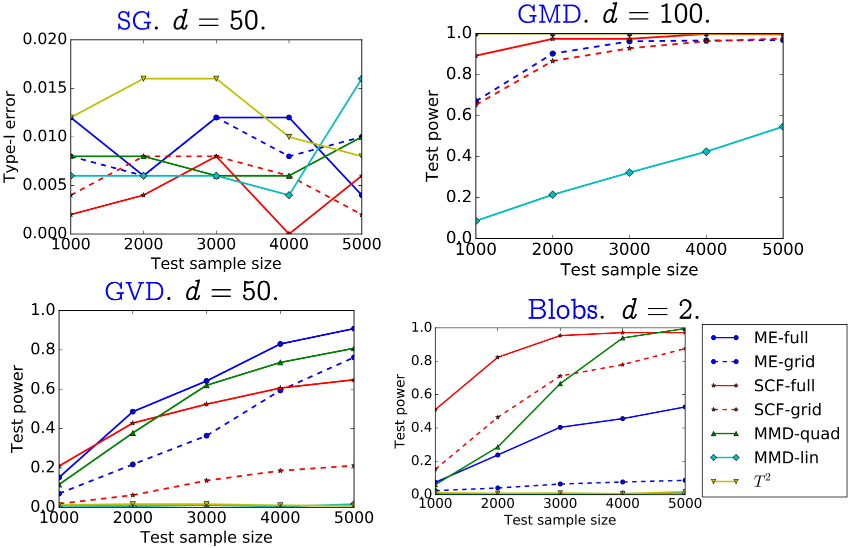
Data	P	Q
1. Same Gaussian (**SG**)	$\mathcal{N}(0_d, I_d)$	$\mathcal{N}(0_d, I_d)$
2. Gauss. mean difference (**GMD**)	$\mathcal{N}(0_d, I_d)$	$\mathcal{N}((1, 0, \ldots, 0)^\top, I_d)$
3. Gauss. variance difference (**GVD**)	$\mathcal{N}(0_d, I_d)$	$\mathcal{N}(0_d, \mathrm{diag}(2, 1, \ldots, 1))$
4. **Blobs** (4×4 grid of Gaussian blobs)		

P :


Q :


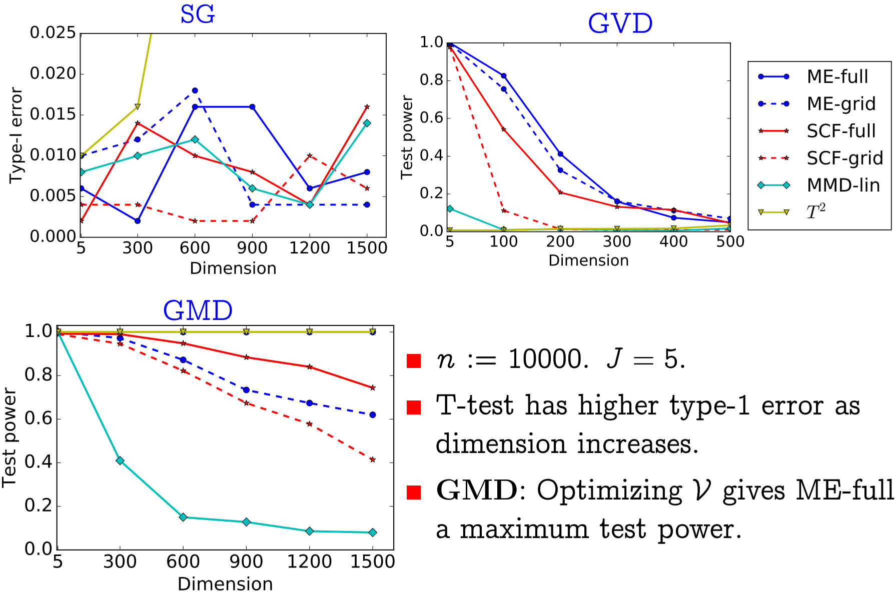
- H_0 is true in SG.
- H_1 is true in others.

Rejection Rate vs. Sample Size



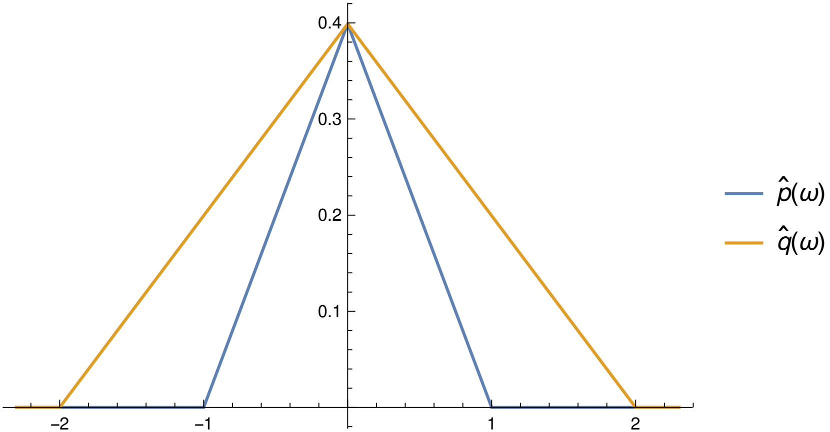
- $J = 5$. Gaussian kernel.
- Right level of type-1 error. Optimizing $\mathcal{V}, \sigma^2$ helps.

Rejection Rate vs. Data Dimension



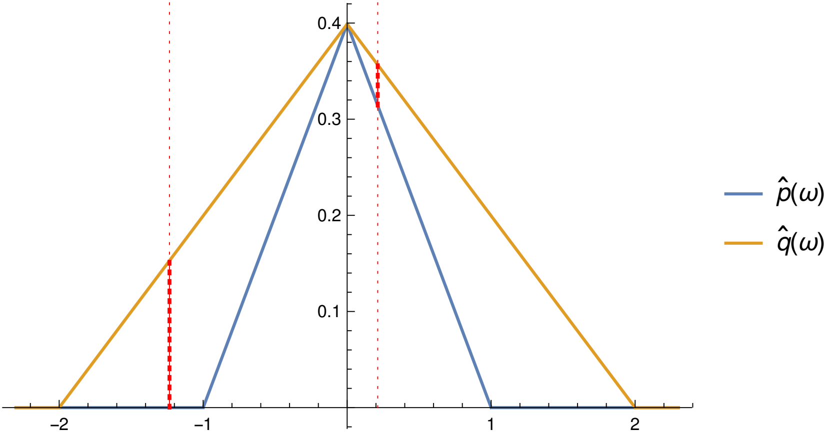
- $n := 10000$. $J = 5$.
- T-test has higher type-1 error as dimension increases.
- **GMD**: Optimizing $\mathcal{V}$ gives ME-full a maximum test power.

Test with smooth characteristic functions (Chwialkowski e



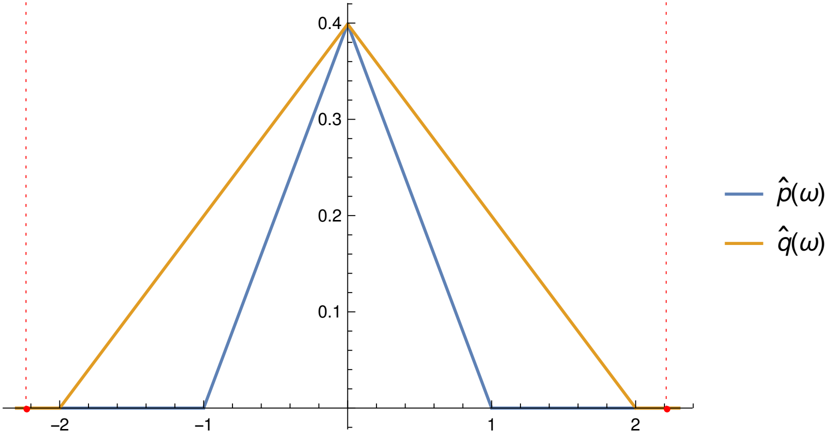
- $\hat{p}(\omega)$, $\hat{q}(\omega)$ are characteristic functions of P, Q.

Illustration: SCF test



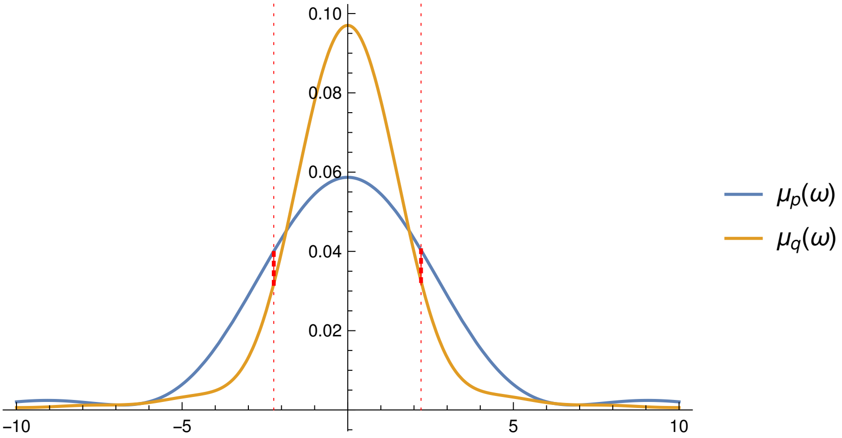
- Checking the difference at finite locations may work.

Illustration: SCF test



- It may also fail if locations are poorly chosen.

Illustration: SCF test



- Smooth the characteristic functions.
- Theoretically, any locations will reveal the difference.

SCF test (Chwialkowski et al., 2015)

- Test based on smooth characteristic functions (SCF) ϕ_P.
- Characteristic function of P is $\hat{p}(\mathbf{w}) := \mathbb{E}_{\mathbf{x}\sim P} \exp(i\mathbf{w}^\top \mathbf{x})$.
- Convolve with an analytic smoothing kernel $l(a) = \exp\left(-\frac{\|a\|^2}{2\sigma^2}\right)$

$$\phi_P(\mathbf{w}) = \int_{\mathbb{R}^d} \hat{p}(\mathbf{w}) l(\mathbf{v}-\mathbf{w})\, d\mathbf{w} \overset{\text{(algebra)}}{=} \int_{\mathbb{R}^d} \exp(i\mathbf{v}^\top \mathbf{x}) \hat{l}(\mathbf{x})\, dP(\mathbf{x}),$$

  where $\hat{l}$ = inverse Fourier transform of l.
- Test statistic: $d^2_{\phi,J}(P,Q) = \frac{1}{J}\sum_{j=1}^{J} \left(\phi_P(\mathbf{v}_j) - \phi_Q(\mathbf{v}_j)\right)^2$.
- $\hat{d}^2_{\phi,J}$ uses
  $\hat{\phi}_P(\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} \exp\left(i\mathbf{v}^\top \mathbf{x}_i\right) \hat{l}(\mathbf{x}_i)$.
- $\mathbf{z}_i :=$

$$\begin{pmatrix} \vdots \\ \hat{l}(\mathbf{x}_i)\sin(\mathbf{x}_i^\top \mathbf{v}_j) - \hat{l}(\mathbf{y}_i)\sin(\mathbf{y}_i^\top \mathbf{v}_j) \\ \hat{l}(\mathbf{x}_i)\cos(\mathbf{x}_i^\top \mathbf{v}_j) - \hat{l}(\mathbf{y}_i)\cos(\mathbf{y}_i^\top \mathbf{v}_j) \\ \vdots \end{pmatrix}.$$

Statistic $\hat{\lambda}_n = n\bar{\mathbf{z}}_n \left(\mathbf{S} + \gamma_n\right)^{-1} \bar{\mathbf{z}}_n$

References I