Learning features to compare distributions

Arthur Gretton

Gatsby Computational Neuroscience Unit,
University College London

NIPS 2016 Workshop on Adversarial Learning,
Barcelona Spain

Goal of this talk

- **Have:** Two collections of samples X, Y from unknown distributions P and Q.
- **Goal:** Learn distinguishing features that indicate how P and Q differ.

$$\sim P$$

$$\sim Q$$

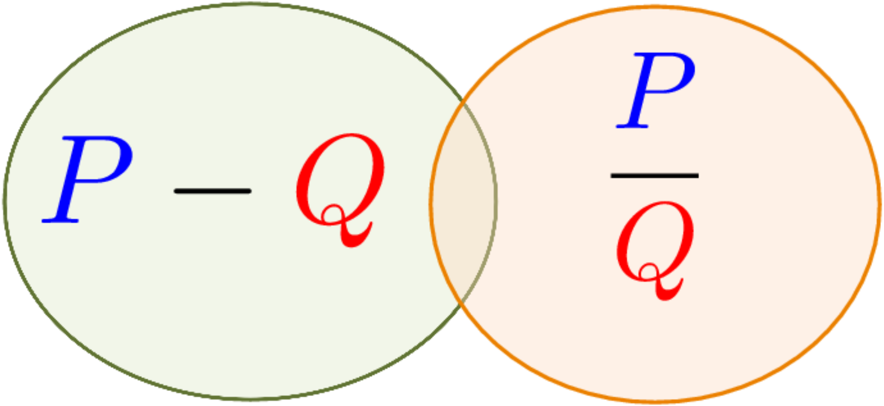
Goal of this talk

- **Have:** Two collections of samples X, Y from unknown distributions P and Q.
- **Goal:** Learn distinguishing features that indicate how P and Q differ.

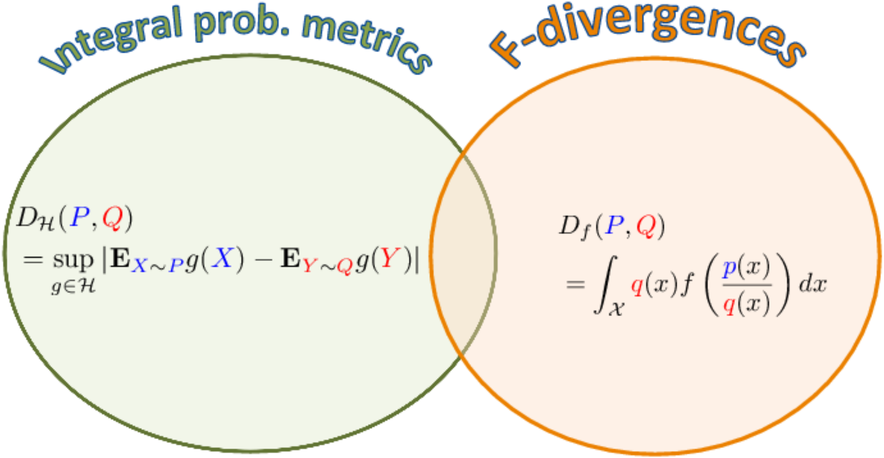
$$\sim P$$

$$\sim Q$$

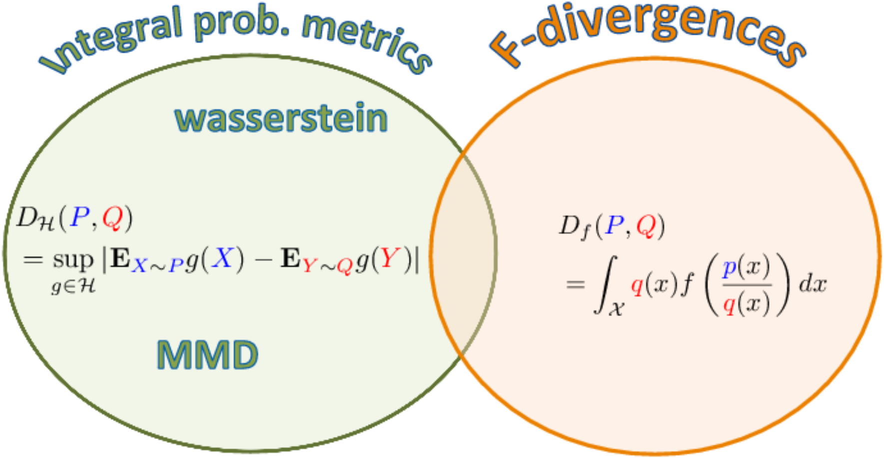
Divergences



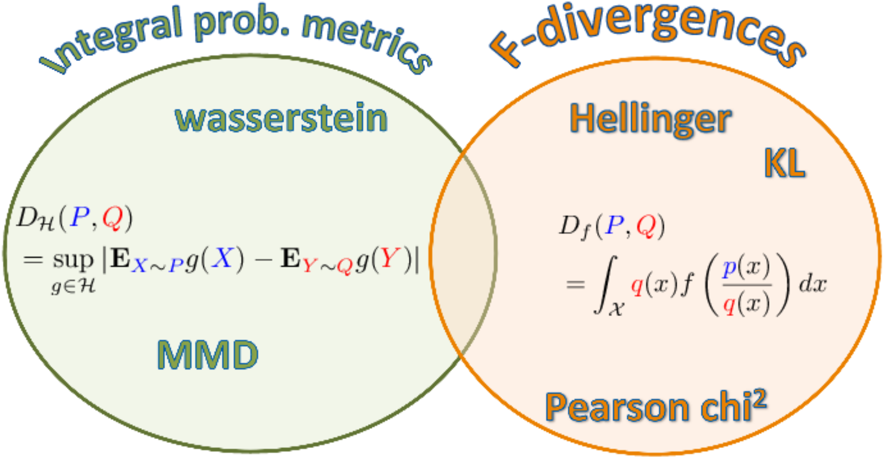
Divergences



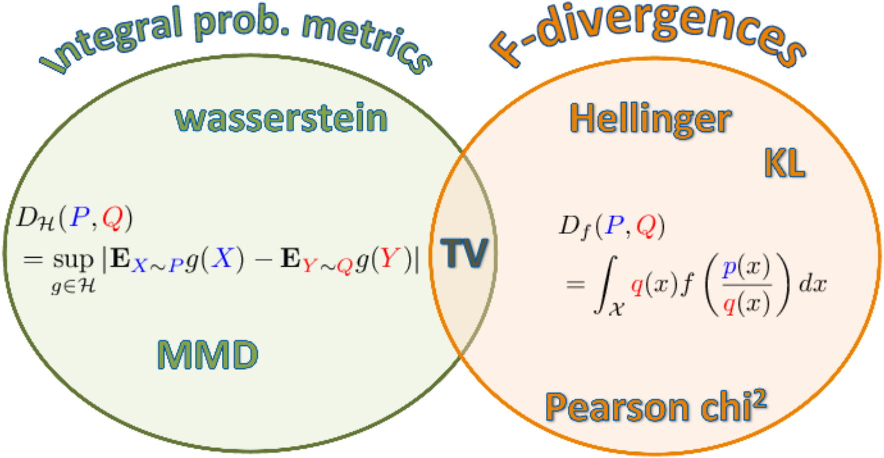
Divergences



Divergences



Divergences



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

Overview

The Maximum mean discrepancy:

- How to compute and interpret the MMD
- How to train the MMD
- Application to troubleshooting GANs

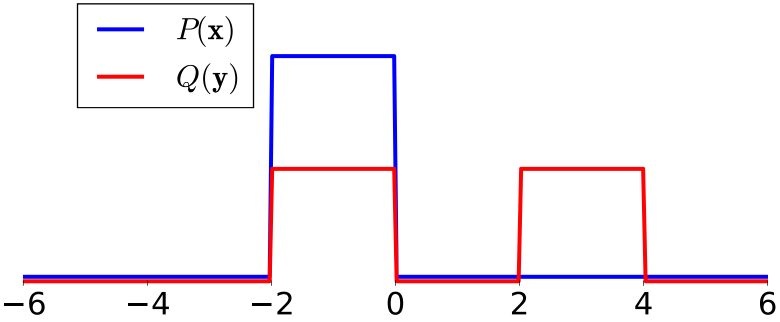
The ME test statistic:

- Informative, linear time features for comparing distributions
- How to learn these features

TL;DR: Variance matters.

The maximum mean discrepancy

Are P and Q different?

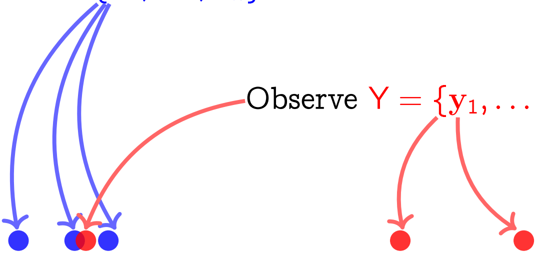


Maximum mean discrepancy (on sample)

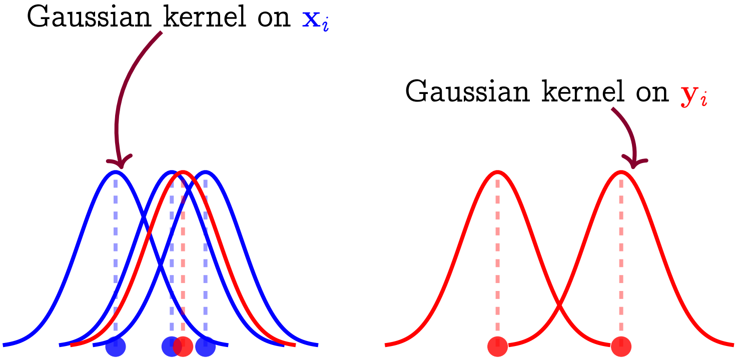
Maximum mean discrepancy (on sample)

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

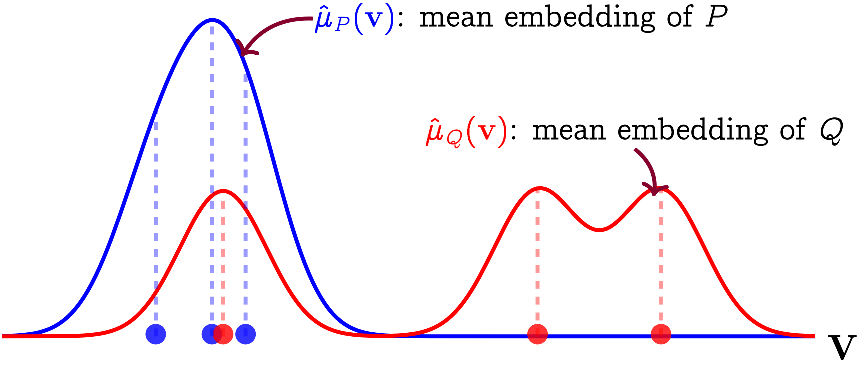
Observe $\mathsf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} \sim Q$



Maximum mean discrepancy (on sample)

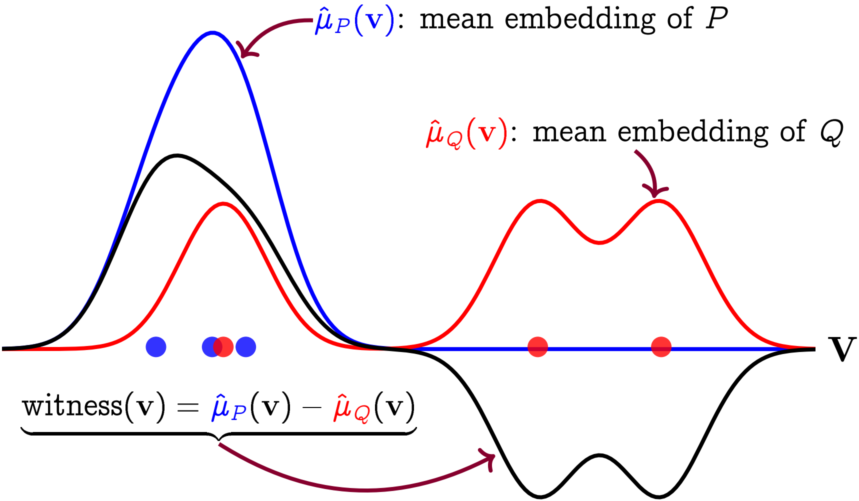


Maximum mean discrepancy (on sample)

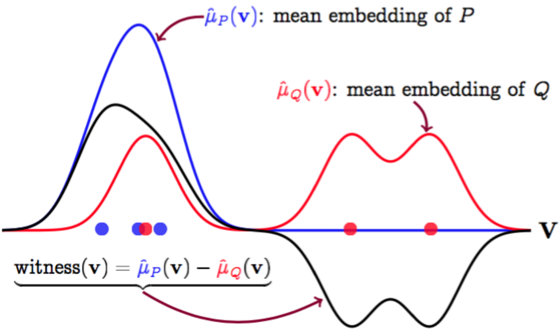


$$\hat{\mu}_P(\mathbf{v}) := \frac{1}{m}\sum_{i=1}^{m} k(x_i, v)$$

Maximum mean discrepancy (on sample)



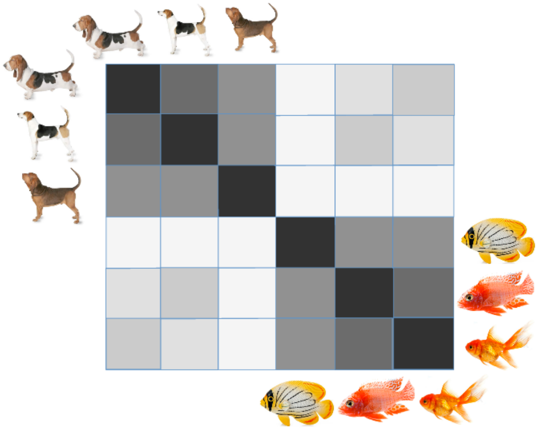
Maximum mean discrepancy (on sample)



$$
\begin{aligned}
\widehat{MMD}^2 &= \|\text{witness}(\mathbf{v})\|_{\mathcal{F}}^2 \\
&= \frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\mathrm{y}_i, \mathrm{y}_j) \\
&\quad - \frac{2}{n^2} \sum_{i,j} k(x_i, \mathrm{y}_j)
\end{aligned}
$$

Overview

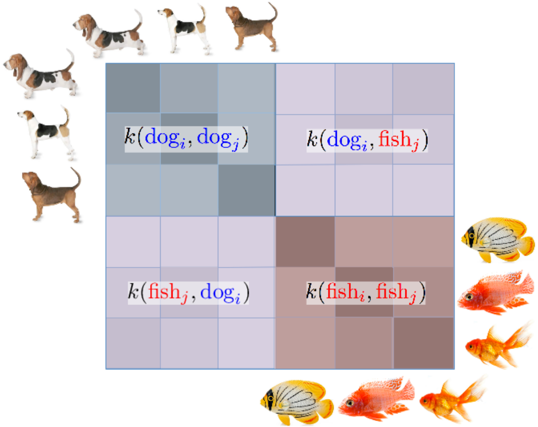
- Dogs ($= P$) and fish ($= Q$) example revisited
- Each entry is one of $k(\text{dog}_i, \text{dog}_j)$, $k(\text{dog}_i, \text{fish}_j)$, or $k(\text{fish}_i, \text{fish}_j)$



Overview

The maximum mean discrepancy:

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



Asymptotics of MMD

- The MMD:

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

but how to choose the kernel?

Asymptotics of MMD

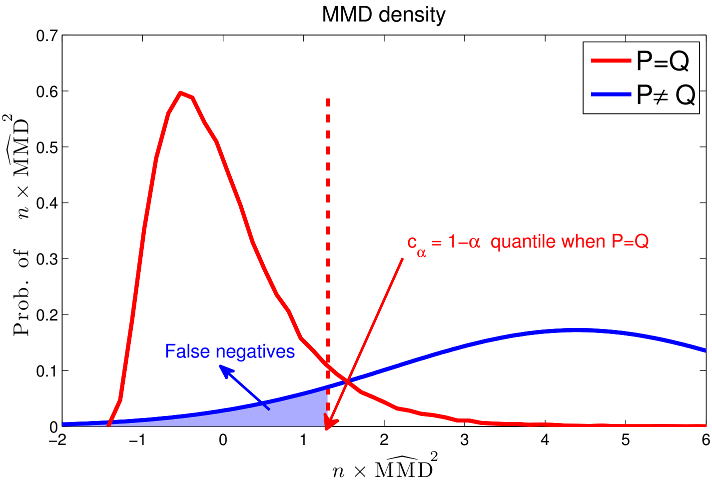
- The MMD:

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

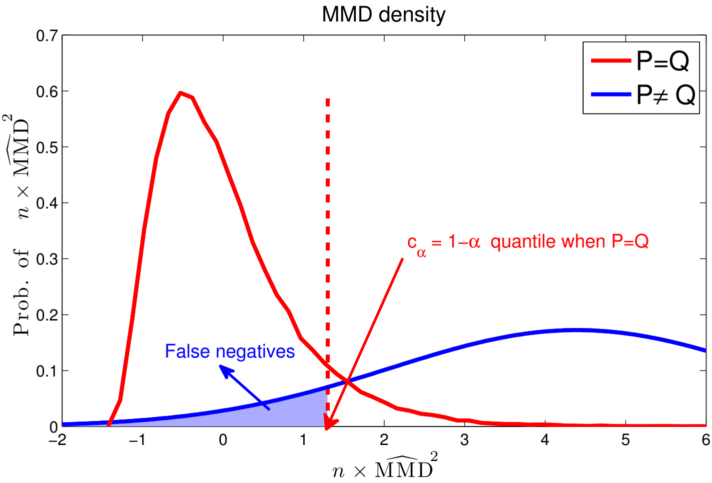
but how to choose the kernel?

- Perspective from statistical hypothesis testing:
  - When $P = Q$ then $\widehat{MMD}^2$ "close to zero".
  - When $P \neq Q$ then $\widehat{MMD}^2$ "far from zero"
- Threshold c_α for $\widehat{MMD}^2$ gives false positive rate α

A statistical test

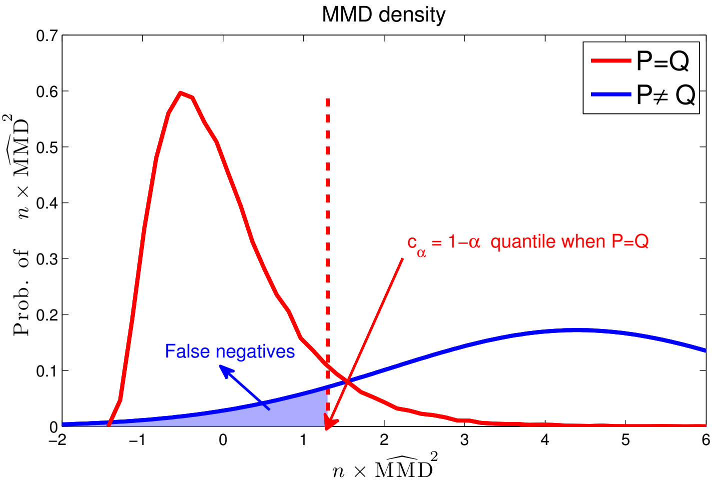


A statistical test



Best kernel gives lowest false negative rate (=highest power)

A statistical test



Best kernel gives lowest false negative rate (=highest power)

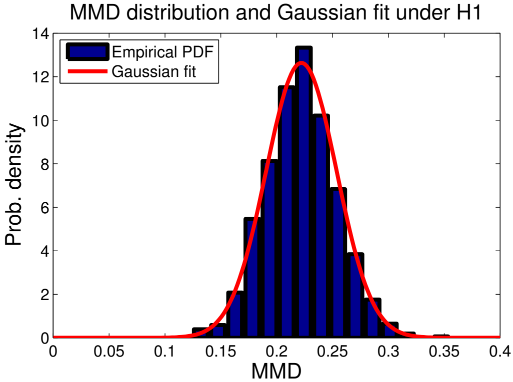
.... but can you train for this?

Asymptotics of MMD

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

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

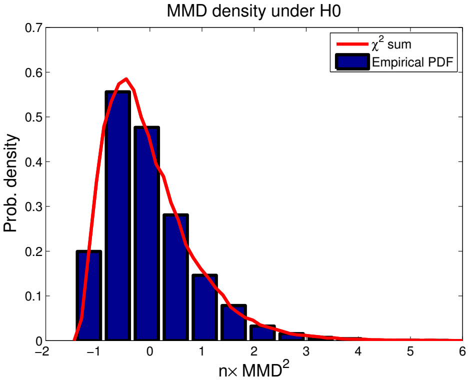
where $\text{MMD}(P, Q)$ is population MMD, and $V_n(P, Q) = O\left(n^{-1}\right)$.



Asymptotics of MMD

Where $P = Q$, statistic has asymptotic distribution

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



where

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

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

Optimizing test power

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

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

Optimizing test power

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

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

where

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

Optimizing test power

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

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

$$\to 1 - \Phi\left(\underbrace{\frac{c_\alpha}{n\sqrt{V_n(P,Q)}}}_{O(n^{-3/2})} - \underbrace{\frac{\mathrm{MMD}^2(P,Q)}{\sqrt{V_n(P,Q)}}}_{O(n^{-1/2})}\right)$$

First term asymptotically negligible!

Optimizing test power

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

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

$$\to 1 - \Phi\left(\frac{c_\alpha}{n\sqrt{V_n(P,Q)}} - \frac{\text{MMD}^2(P,Q)}{\sqrt{V_n(P,Q)}}\right)$$

To maximize test power, maximize

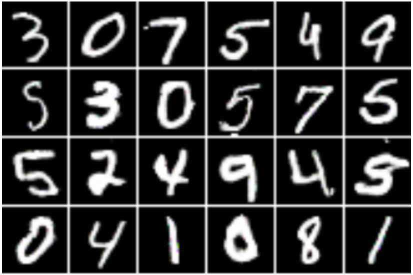
$$\frac{\text{MMD}^2(P,Q)}{\sqrt{V_n(P,Q)}}$$

(Sutherland, Tung, Strathmann, De, Ramdas, Smola, G., in review for ICLR 2017)

Code: github.com/dougalsutherland/opt-mmd

Troubleshooting for generative adversarial networks

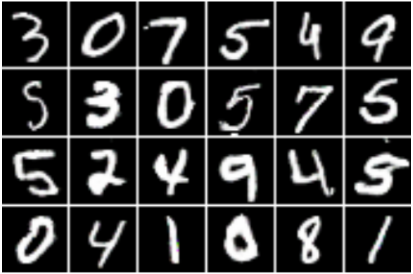
MNIST samples



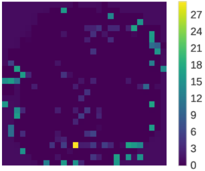
Samples from a GAN

Troubleshooting for generative adversarial networks

MNIST samples



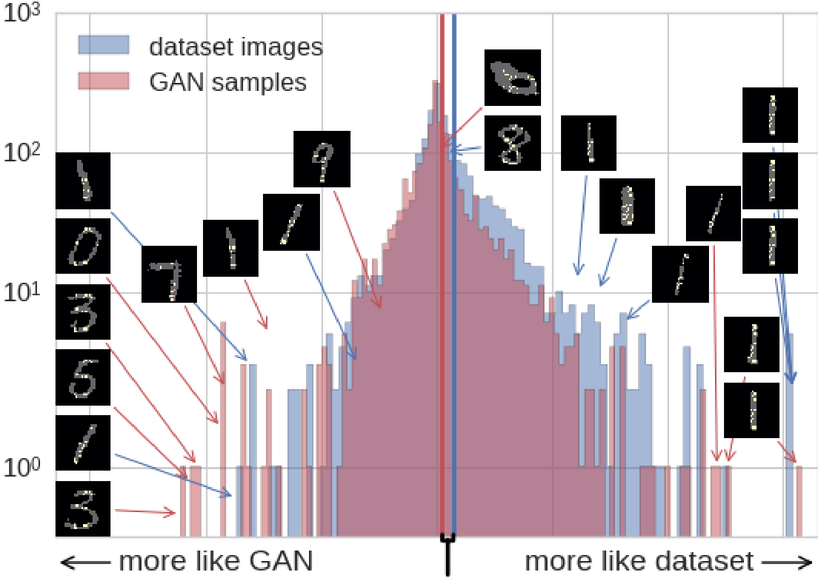
Samples from a GAN



ARD map

- Power for **optimzed ARD kernel**: 1.00 at $\alpha = 0.01$
- Power for optimized RBF kernel: 0.57 at $\alpha = 0.01$

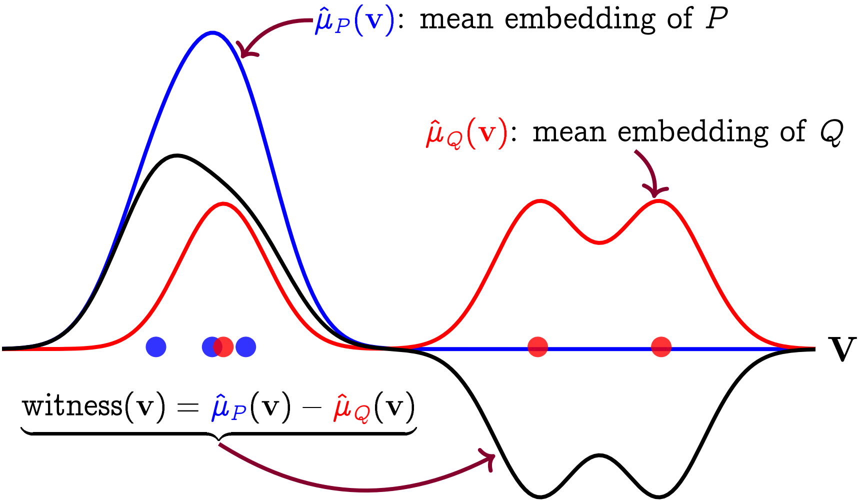
Benchmarking generative adversarial networks



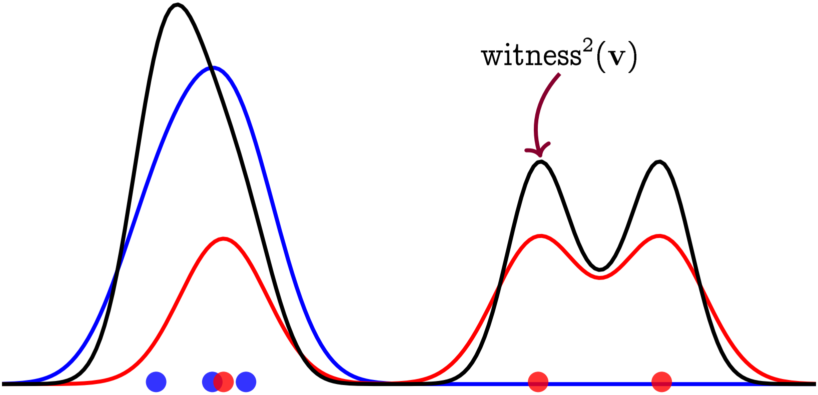
$\mathrm{MMD}^2 = 0.0001$

The ME statistic and test

Distinguishing Feature(s)

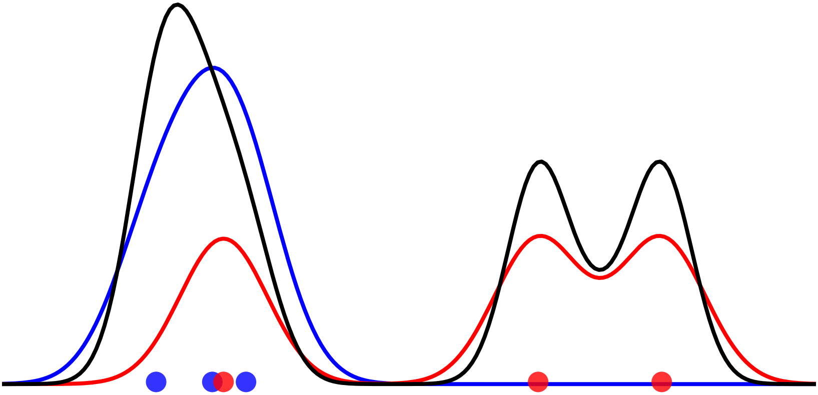


Distinguishing Feature(s)



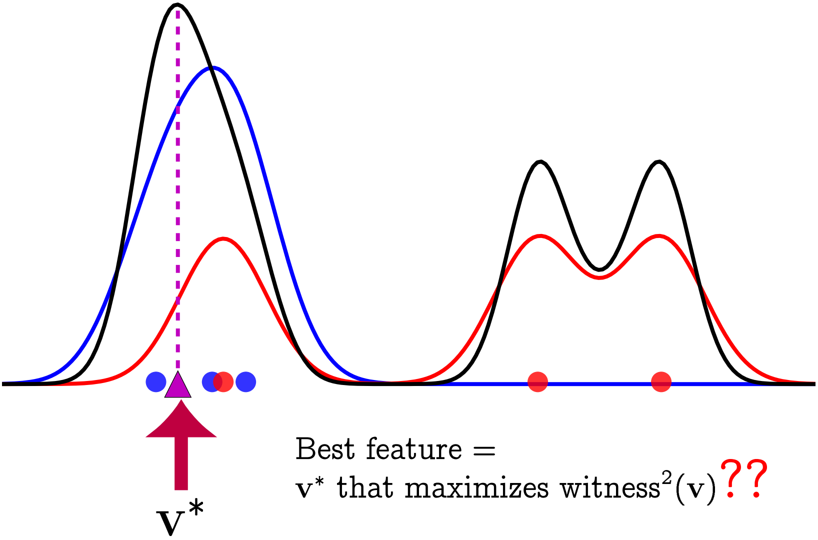
Take square of witness (only worry about amplitude)

Distinguishing Feature(s)



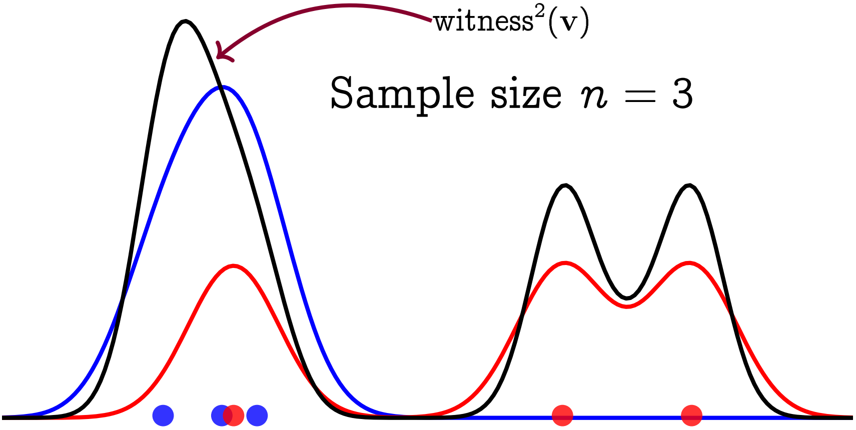
- New test statistic: witness^2 at a single $\mathbf{v}^*$;
- Linear time in number n of samples
- ....but how to choose best feature $\mathbf{v}^*$?

Distinguishing Feature(s)



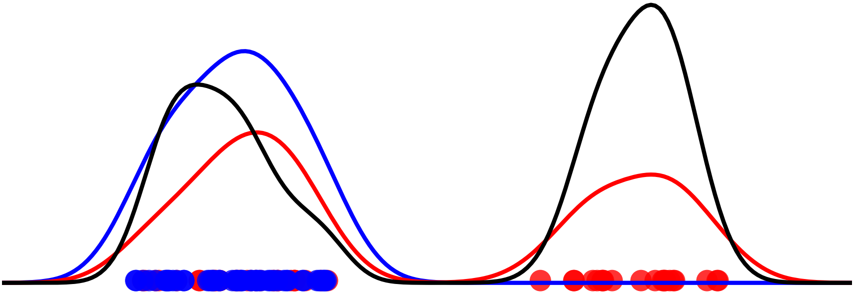
Best feature =
$\mathbf{v}^*$ that maximizes witness$^2(\mathbf{v})$??

Distinguishing Feature(s)



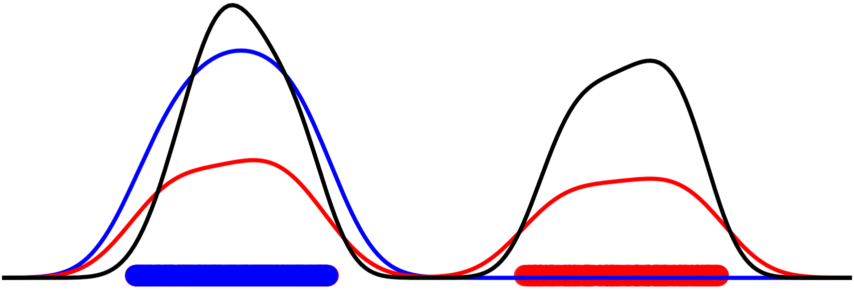
Distinguishing Feature(s)

Sample size $n = 50$

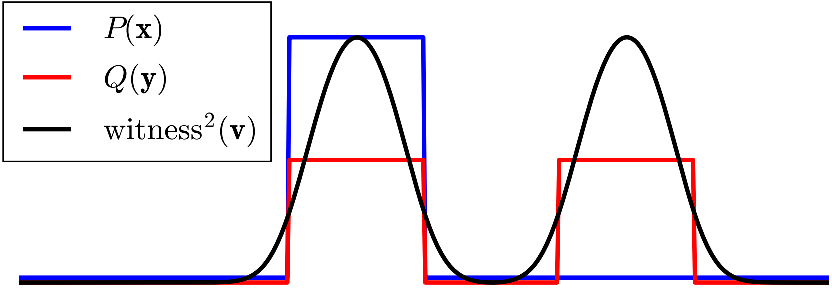


Distinguishing Feature(s)

Sample size $n = 500$

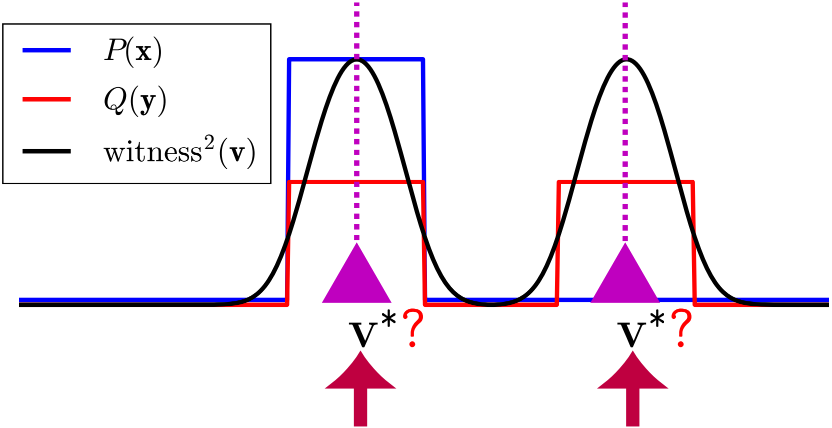


Distinguishing Feature(s)



Population witness^2 function

Distinguishing Feature(s)



Variance of witness function

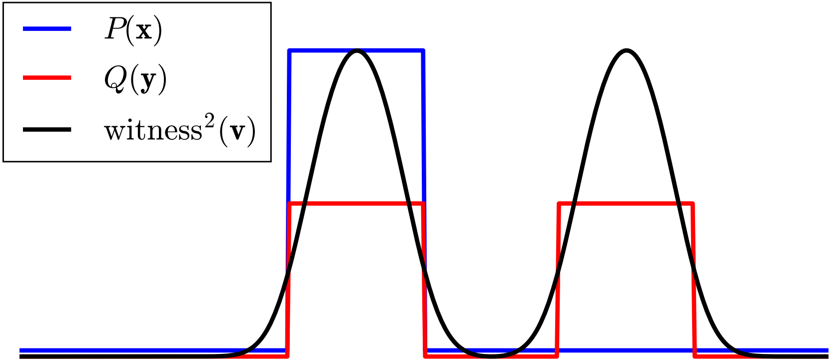
- Variance at $\mathbf{v}$ = variance of X at $\mathbf{v}$ + variance of Y at $\mathbf{v}$.
- ME Statistic: $\hat{\lambda}_n(\mathbf{v}) := n \frac{\text{witness}^2(\mathbf{v})}{\text{variance of } \mathbf{v}}$.

Variance of witness function

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

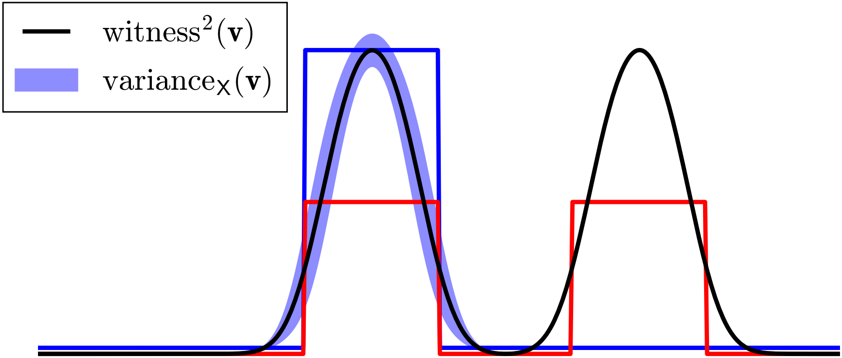
Variance of witness function

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



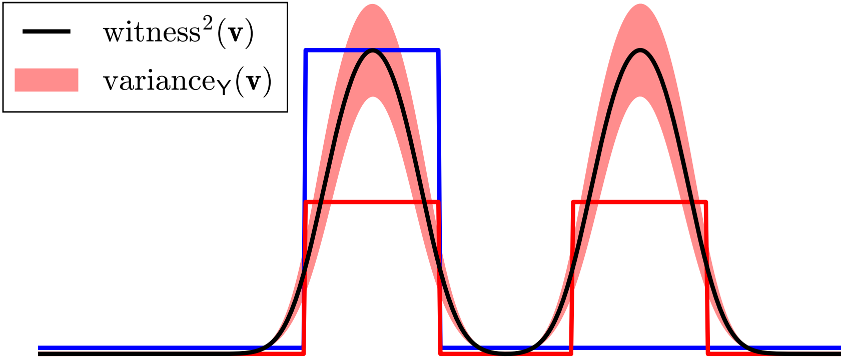
Variance of witness function

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



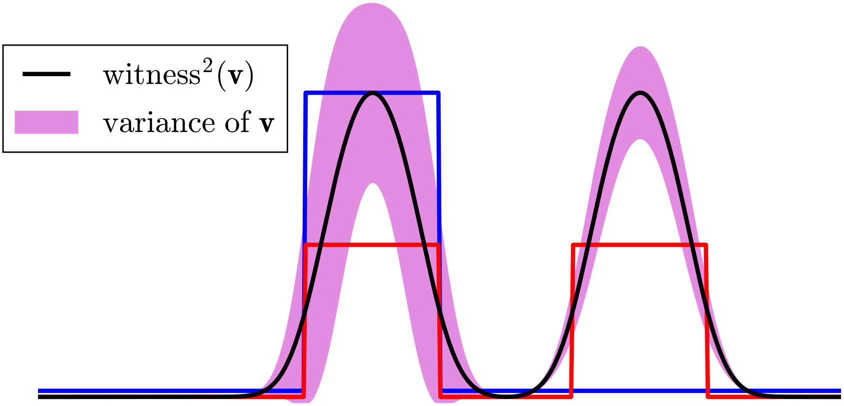
Variance of witness function

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



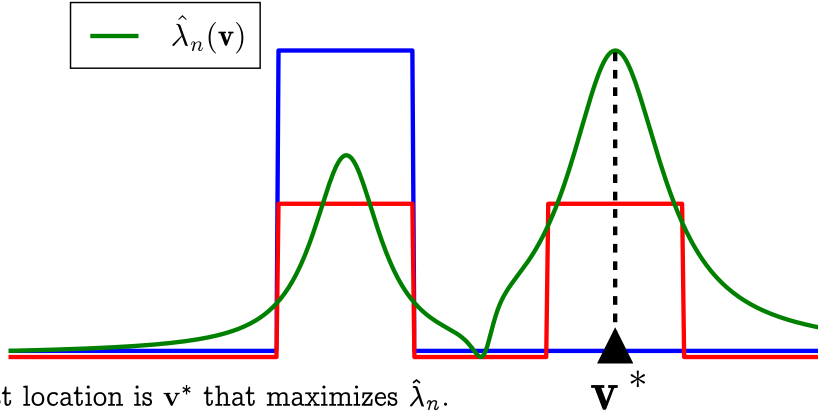
Variance of witness function

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



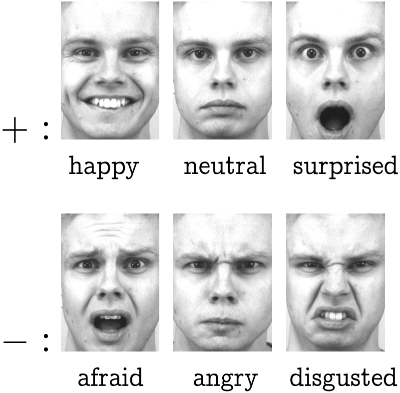
Variance of witness function

- Variance at $\mathbf{v}$ = variance of X at $\mathbf{v}$ + variance of Y at $\mathbf{v}$.
- 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 $\hat{\lambda}_n$.
- Improve performance using multiple locations $\{\mathbf{v}_j^*\}_{j=1}^J$

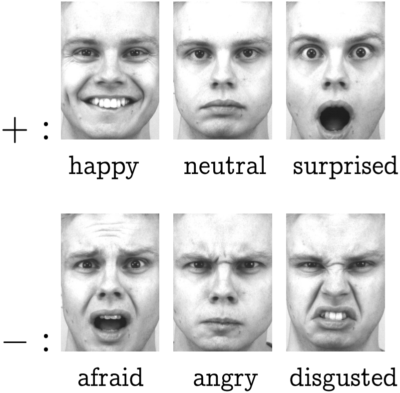
Distinguishing Positive/Negative Emotions

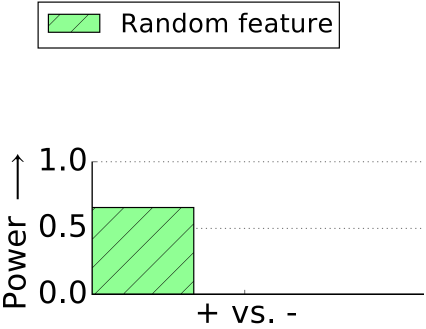


- 35 females and 35 males (Lundqvist et al., 1998).
- $48 \times 34 = 1632$ dimensions. Pixel features.
- Sample size: 402.

- The proposed test achieves **maximum test power** in **time** $O(n)$.
- **Informative features:** differences at the nose, and smile lines.

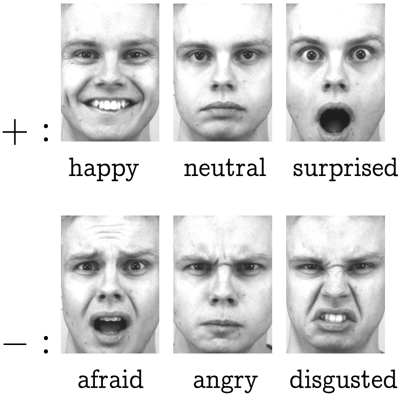
Distinguishing Positive/Negative Emotions

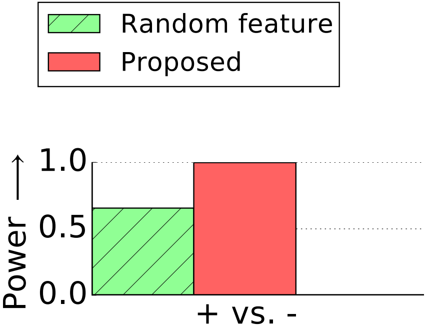




- The proposed test achieves **maximum test power** in **time** $O(n)$.
- **Informative features:** differences at the nose, and smile lines.

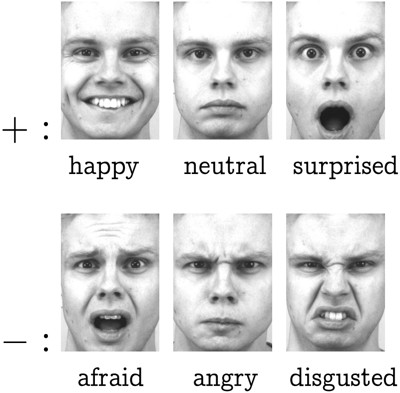
Distinguishing Positive/Negative Emotions

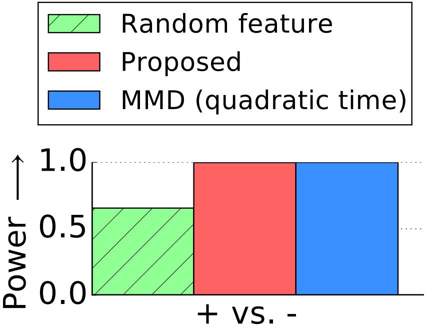




- The proposed test achieves **maximum test power** in **time $O(n)$**.
- Informative features: differences at the nose, and smile lines.

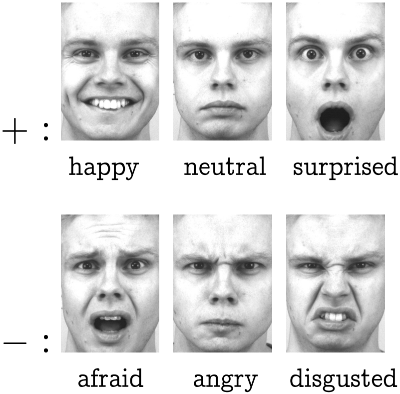
Distinguishing Positive/Negative Emotions

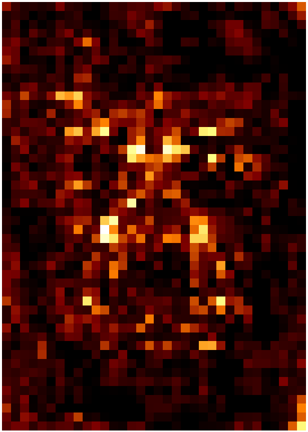




- The proposed test achieves **maximum test power** in **time $O(n)$**.
- Informative features: differences at the nose, and smile lines.

Distinguishing Positive/Negative Emotions

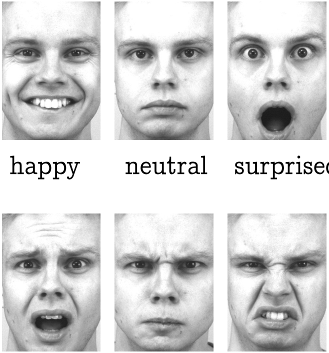




Learned feature

- The proposed test achieves **maximum test power** in **time $O(n)$**.
- **Informative features**: differences at the nose, and smile lines.

Distinguishing Positive/Negative Emotions



+ : happy neutral surprised

− : afraid angry disgusted

Learned feature

- The proposed test achieves **maximum test power** in **time** $O(n)$.
- **Informative features**: differences at the nose, and smile lines.

Code: https://github.com/wittawatj/interpretable-test

Final thoughts

Witness function approaches:

- Diversity of samples:
  - MMD test uses pairwise similarities between all samples
  - ME test uses similarities to J reference features
- Disjoint support of generator/data distributions
  - Witness function is smooth

Other discriminator heuristics:

- Diversity of samples by minibatch heuristic (add as feature distances to neighbour samples) Salimans et al. (2016)
- Disjoint support treated by adding noise to “blur” images Arjovsky and Bottou (2016), Sønderby et al (2016)

Co-authors

Students and postdocs:

- Kacper Chwialkowski (at Voleon)
- Wittawat Jitkrittum
- Heiko Strathmann
- Dougal Sutherland

Collaborators

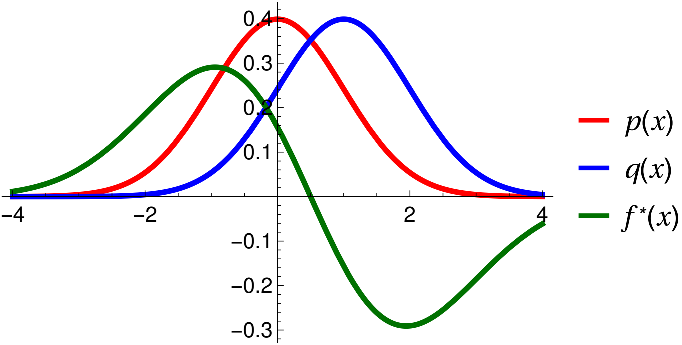
- Kenji Fukumizu
- Krikamol Muandet
- Bernhard Schoelkopf
- Bharath Sriperumbudur
- Zoltan Szabo

Questions?

Testing against a probabilistic model

Statistical model criticism

$$MMD(P, Q) = \|f^*\|^2 = \sup_{\|f\|_{\mathcal{F}} \leq 1} [E_Q f - E_p f]$$



$f^*(x)$ is the witness function

Can we compute MMD with samples from Q and a **model** P?

Problem: usualy can't compute $E_p f$ in closed form.

Stein idea

To get rid of $E_p f$ in

$$\sup_{||f||_{\mathcal{F}} \leq 1} [E_q f - E_p f]$$

we define the **Stein operator**

$$T_p f = \partial_x f + f\left(\partial_x \log p\right)$$

Then

$$E_P T_P f = 0$$

subject to appropriate boundary conditions. (Oates, Girolami, Chopin, 2016)

Maximum Stein Discrepancy

Stein operator

$$T_p f = \partial_x f + f \partial_x (\log p)$$

Maximum Stein Discrepancy (MSD)

$$MSD(p, q, \mathcal{F}) = \sup_{\|g\|_{\mathcal{F}} \leq 1} E_q T_p g - E_p T_p g$$

Maximum Stein Discrepancy

Stein operator

$$T_p f = \partial_x f + f \partial_x (\log p)$$

Maximum Stein Discrepancy (MSD)

$$MSD(p, q, \mathcal{F}) = \sup_{\|g\|_{\mathcal{F}} \leq 1} E_q T_p g - \cancel{E_p T_p g}$$

Maximum Stein Discrepancy

Stein operator

$$T_p f = \partial_x f + f \partial_x (\log p)$$

Maximum Stein Discrepancy (MSD)

$$MSD(p, q, \mathcal{F}) = \sup_{\|g\|_{\mathcal{F}} \leq 1} E_q T_p g - \cancel{E_p T_p g} = \sup_{\|g\|_{\mathcal{F}} \leq 1} E_q T_p g$$

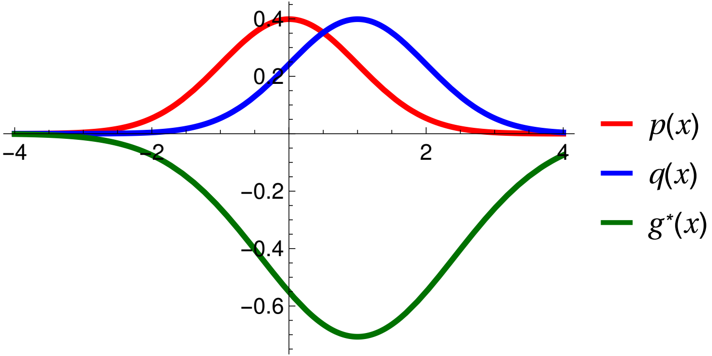
Maximum Stein Discrepancy

Stein operator

$$T_p f = \partial_x f + f \partial_x (\log p)$$

Maximum Stein Discrepancy (MSD)

$$MSD(p, q, \mathcal{F}) = \sup_{\|g\|_{\mathcal{F}} \leq 1} E_q T_p g - \cancel{E_p T_p g} = \sup_{\|g\|_{\mathcal{F}} \leq 1} E_q T_p g$$



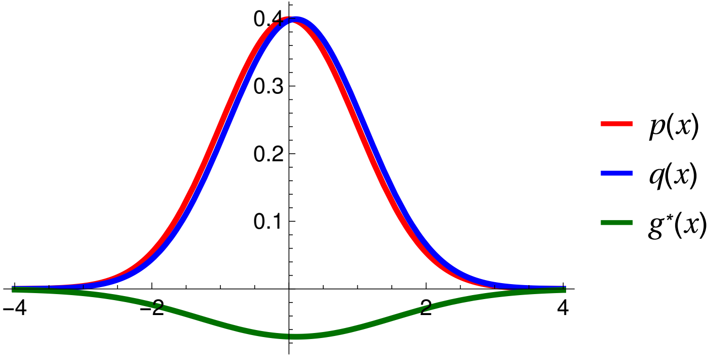
Maximum Stein Discrepancy

Stein operator

$$T_p f = \partial_x f + f \partial_x (\log p)$$

Maximum Stein Discrepancy (MSD)

$$MSD(p, q, \mathcal{F}) = \sup_{\|g\|_{\mathcal{F}} \leq 1} E_q T_p g - \cancel{E_p T_p g} = \sup_{\|g\|_{\mathcal{F}} \leq 1} E_q T_p g$$



Maximum stein discrepancy

Closed-form expression for MSD: given $Z, Z' \sim q$, then (Chwialkowski, Strathmann, G., 2016) (Liu, Lee, Jordan 2016)

$$\text{MSD}(p, q, \mathcal{F}) = E_q h_p(Z, Z')$$

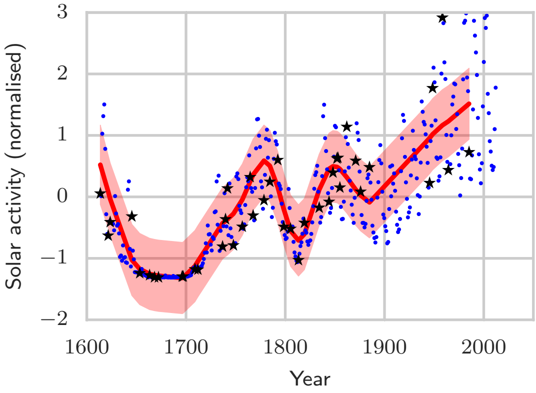
where

$$\begin{aligned} h_p(x,y) := & \, \partial_x \log p(x) \partial_x \log p(y) k(x,y) \\ & + \partial_y \log p(y) \partial_x k(x,y) \\ & + \partial_x \log p(x) \partial_y k(x,y) \\ & + \partial_x \partial_y k(x,y) \end{aligned}$$

and k is RKHS kernel for $\mathcal{F}$

Only depends on kernel and $\partial_x \log p(x)$. Do not need to normalize p, or sample from it.

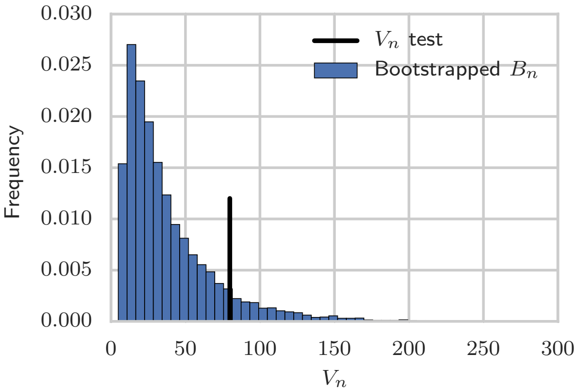
Statistical model criticism



Test the hypothesis that a Gaussian process model, learned from **data** $\star$, is a good fit for the test data (example from Lloyd and Ghahramani, 2015)

Code: https://github.com/karlnapf/kernel_goodness_of_fit

Statistical model criticism



Test the hypothesis that a Gaussian process **model**, learned from **data** $\star$, is a good fit for the test data