Conditional Density Estimation, Kernel Embeddings, and Meta Learning

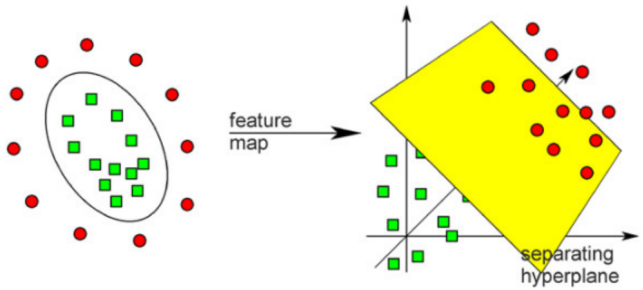
Dino Sejdinovic

Department of Statistics
University of Oxford

Workshop on Functional Inference and Machine Intelligence
17/02/2020

Kernel Trick and Kernel Mean Trick

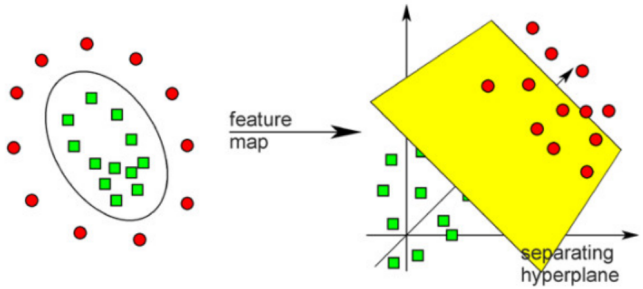
- implicit feature map $x \mapsto k(\cdot, x) \in \mathcal{H}_k$
  replaces $x \mapsto [\phi_1(x), \ldots, \phi_s(x)] \in \mathbb{R}^s$
- $\langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}_k} = k(x, y)$
  inner products readily available
  - nonlinear decision boundaries, nonlinear regression functions, learning on non-Euclidean/structured data



[Cortes & Vapnik, 1995; Schölkopf & Smola, 2001]

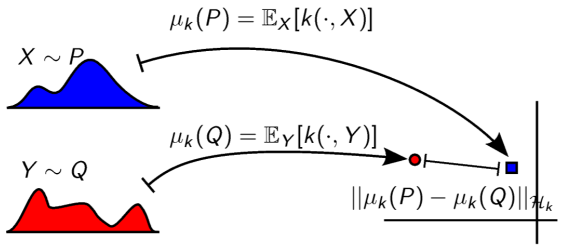
Kernel Trick and Kernel Mean Trick

- implicit feature map $x \mapsto k(\cdot, x) \in \mathcal{H}_k$
  replaces $x \mapsto [\phi_1(x), \dots, \phi_s(x)] \in \mathbb{R}^s$
- $\langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}_k} = k(x, y)$
  inner products readily available
  - nonlinear decision boundaries, nonlinear regression functions, learning on non-Euclidean/structured data



[Cortes & Vapnik, 1995; Schölkopf & Smola, 2001]

- **RKHS embedding**: implicit feature mean
  [Smola et al, 2007; Sriperumbudur et al, 2010; Muandet et al, 2017]
  $P \mapsto \mu_k(P) = \mathbb{E}_{X \sim P} k(\cdot, X) \in \mathcal{H}_k$
  replaces $P \mapsto [\mathbb{E}\phi_1(X), \dots, \mathbb{E}\phi_s(X)] \in \mathbb{R}^s$
- $\langle \mu_k(P), \mu_k(Q) \rangle_{\mathcal{H}_k} = \mathbb{E}_{X \sim P, Y \sim Q} k(X, Y)$
  inner products easy to estimate
  - nonparametric two-sample, independence, conditional independence, interaction testing, learning on distributions

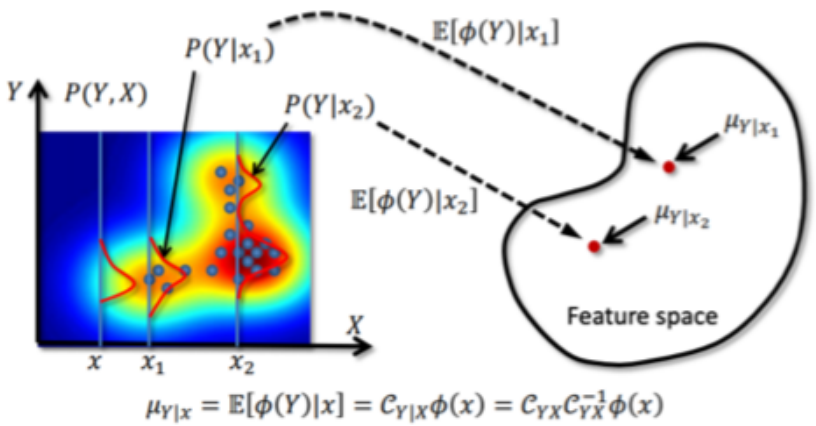


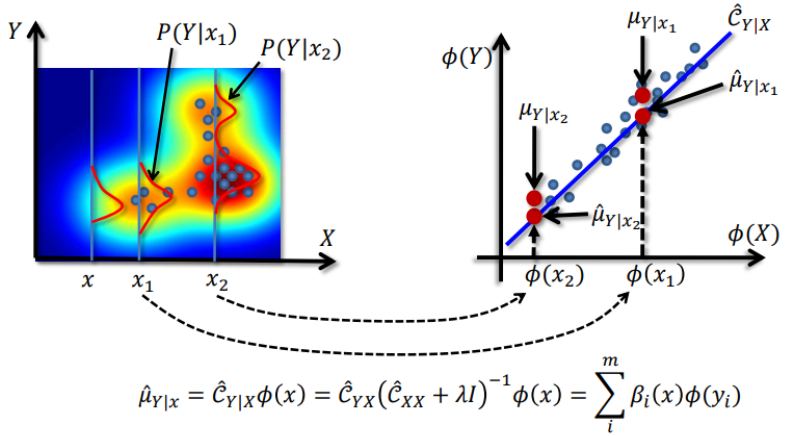
[Gretton et al, 2005; Gretton et al, 2006; Fukumizu et al, 2007; DS et al, 2013; Muandet et al, 2012; Szabo et al, 2015]

Conditional Mean Embeddings

Consider a joint distribution P_{XY} over the random variables (X, Y) taking values in $\mathcal{X} \times \mathcal{Y}$. The conditional mean embedding (CME) of $Y|X = x$ is defined as:

$$\mu_{Y|X=x} := \mathbb{E}_{Y|X=x}[k_y(\cdot, Y)] = \int_{\mathcal{Y}} k_y(\cdot, y) dP(y|x) \in \mathcal{H}_{k_y}$$





To model conditional embeddings as functions of x, we associate them with a linear operator $\mathcal{C}_{Y|X} : \mathcal{H}_{k_x} \to \mathcal{H}_{k_y}$, which satisfies

$$\mu_{Y|X=x} = \mathcal{C}_{Y|X} k_x(\cdot, x).$$

This is essentially feature-to-feature (RKHS-to-RKHS) penalized regression.

Review of CMEs in [Song et al, 2013].

Conditional Mean Embedding Operator (CMEO)

The conditional mean embedding can be associated with the operator $\mathcal{C}_{Y|X} : \mathcal{H}_x \to \mathcal{H}_y$, which satisfies

$$\mu_{Y|X=x} = \mathcal{C}_{Y|X} k_x(\cdot, x).$$

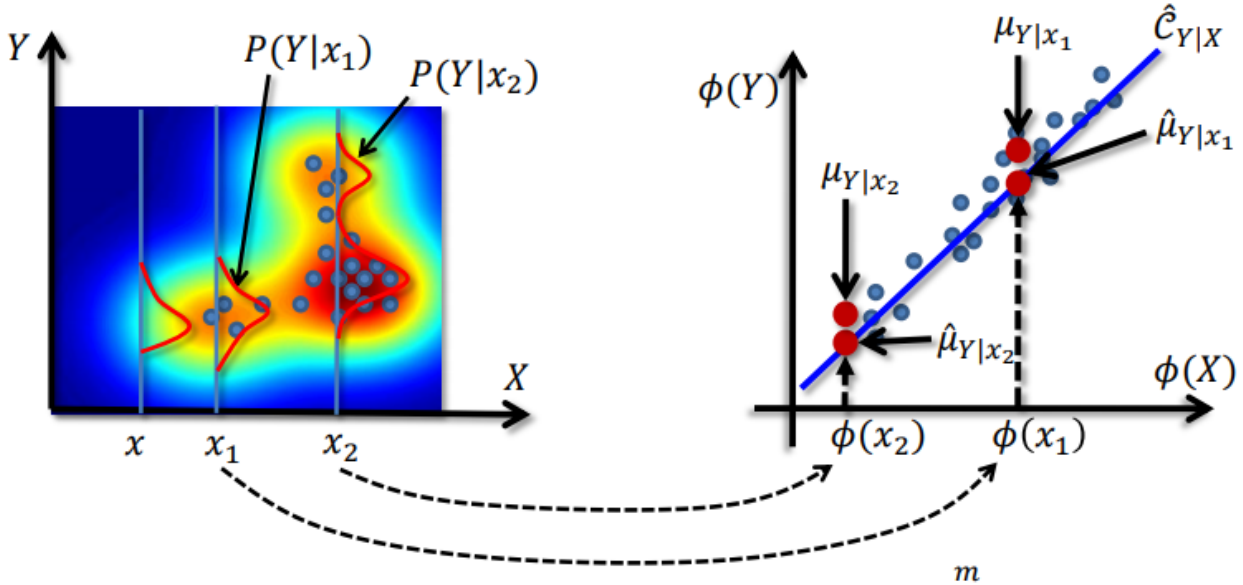
We can write $\mathcal{C}_{Y|X} := \mathcal{C}_{YX}\mathcal{C}_{XX}^{-1}$ where $\mathcal{C}_{YX} := \mathbb{E}_{Y,X}[k_y(\cdot, Y) \otimes k_x(\cdot, X)]$ and $\mathcal{C}_{XX} := \mathbb{E}_{X,X}[k_x(\cdot, X) \otimes k_x(\cdot, X)]$.

Estimation of CMEO

Considering feature maps ϕ_x and ϕ_y, the finite sample estimator of $\mathcal{C}_{Y|X}$ based on dataset $\{(x_i, y_i)\}_{i=1}^n$ is given by feature-to-feature regression coefficients:

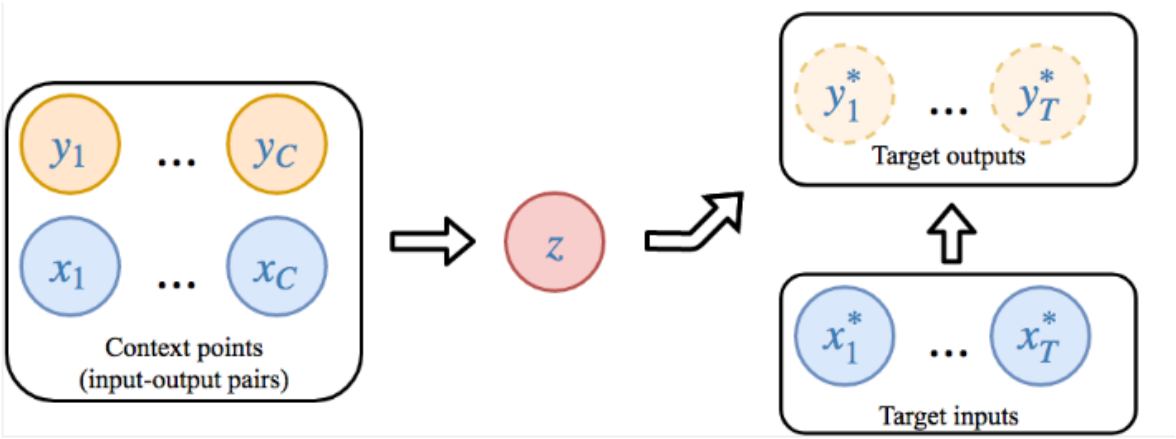
$$\widehat{\mathcal{C}}_{Y|X} = \Phi_y (K_{xx} + \lambda I)^{-1} \Phi_x^\top,$$

where $\Phi_y := (\phi_y(y_1), \ldots, \phi_y(y_n))$ and $\Phi_x := (\phi_x(x_1), \ldots, \phi_x(x_n))$ are the feature matrices, $K_{xx} := \Phi_x \Phi_x^\top$ is the kernel matrix with entries $[K_{xx}]_{i,j} = k_x(x_i, x_j) := \langle \phi_x(x_i), \phi_x(x_j) \rangle$, and $\lambda > 0$ is a regularization parameter of feature-to-feature regression.



$$\hat{\mu}_{Y|x} = \hat{\mathcal{C}}_{Y|X}\phi(x) = \hat{\mathcal{C}}_{YX}\left(\hat{\mathcal{C}}_{XX} + \lambda I\right)^{-1}\phi(x) = \sum_i^m \beta_i(x)\phi(y_i)$$

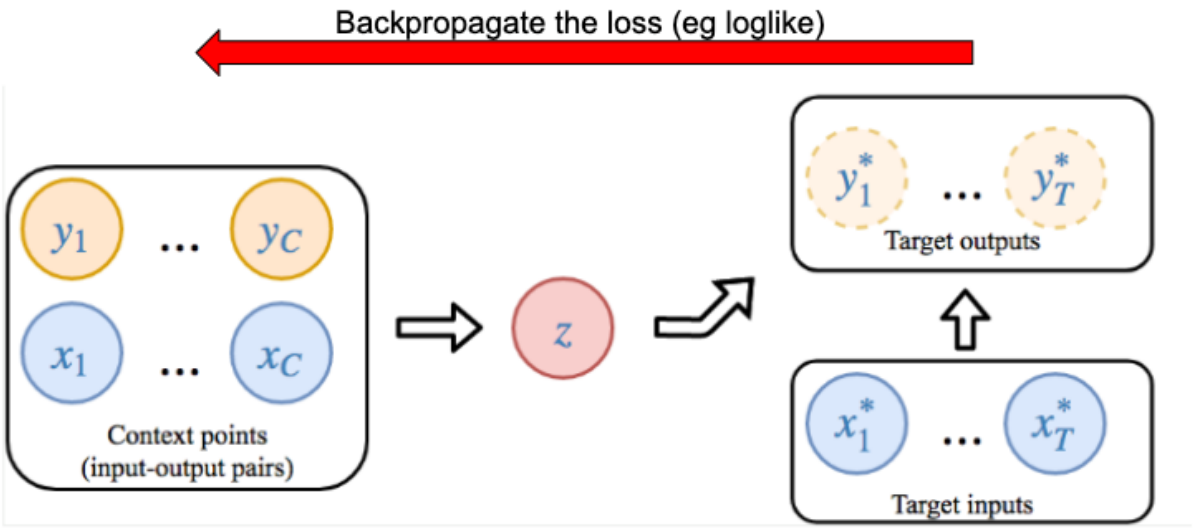
Probabilistic Meta Learning Framework [Garnelo et al 18]



- Let $\mathcal{T} = \{T_1, \ldots, T_L\}$ be the set of L tasks, each divided into context $\mathcal{D}_c^l = \{(x_i^{l,c}, y_i^{l,c})\}$ and target data $\mathcal{D}_t^l = \{(x_i^{l,t}, y_i^{l,t})\}$
- Context set is to extract the meta information, encoded as the "task embedding"
- Target set is to test how well the information was extracted by compute the loss on the target set.
- During testing time we only have context set and are asked to predict on any new x^*

[Thrun and Pratt, 1998; Ravi and Larochelle, 2016; Santoro et al., 2016]

Probabilistic Meta Learning Framework [Garnelo et al 18]

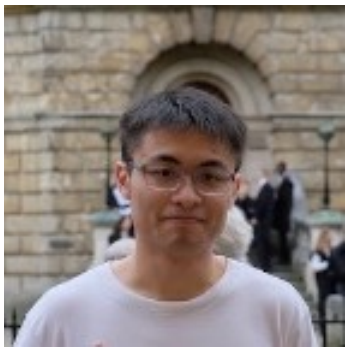
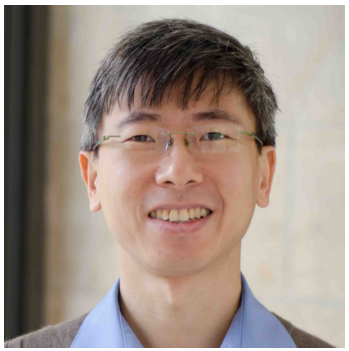


- Let $\mathcal{T} = \{T_1, \ldots, T_L\}$ be the set of L tasks, each divided into context $\mathcal{D}_c^l = \{(x_i^{l,c}, y_i^{l,c})\}$ and target data $\mathcal{D}_t^l = \{(x_i^{l,t}, y_i^{l,t})\}$
- Context set is to extract the meta information, encoded as the "task embedding"
- Target set is to test how well the information was extracted by compute the loss on the target set.
- During testing time we only have context set and are asked to predict on any new x^*

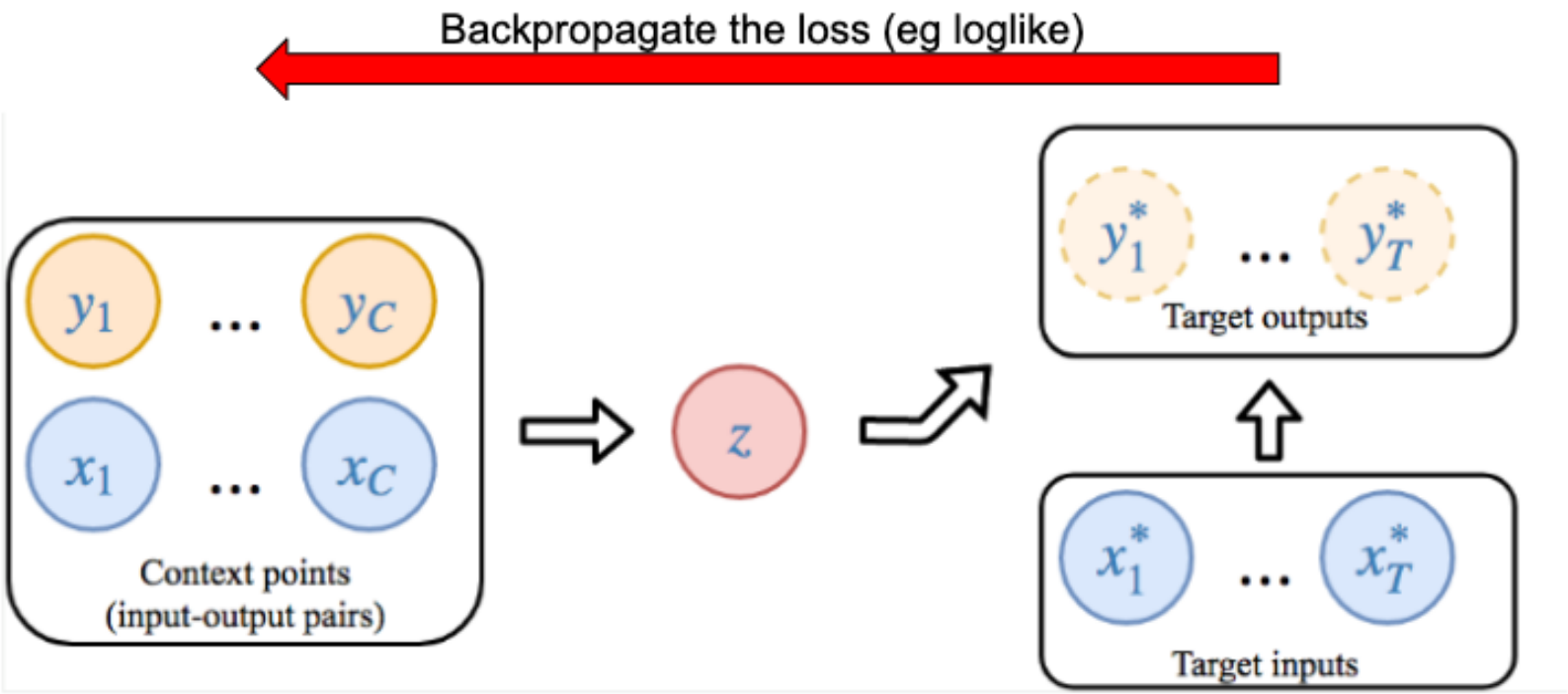
[Thrun and Pratt, 1998; Ravi and Larochelle, 2016; Santoro et al., 2016]

Kernel Embeddings for Meta Learning

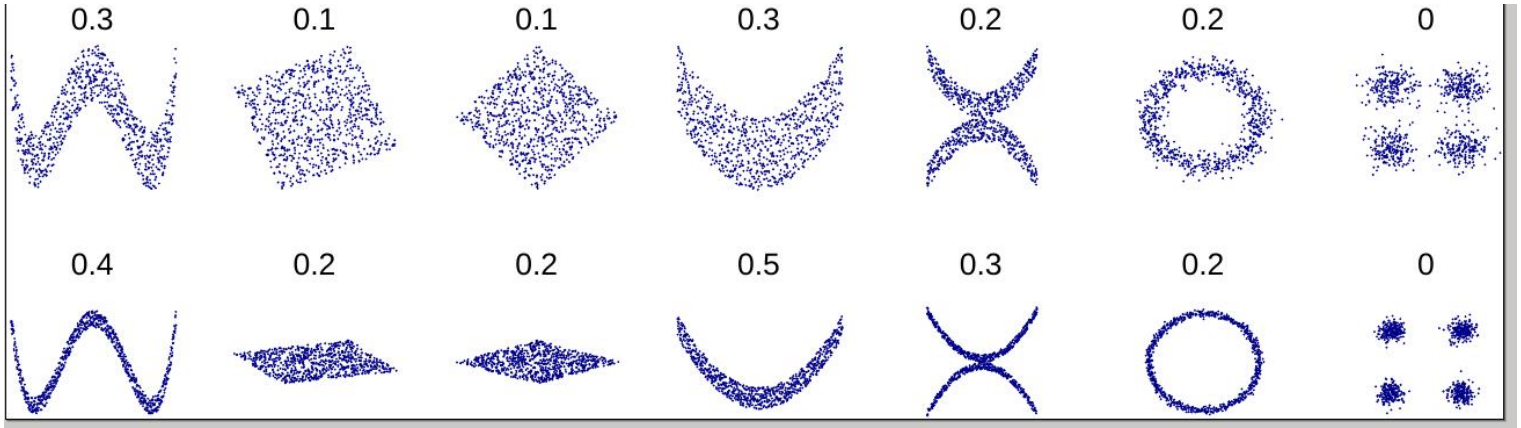
- Jean-Francois Ton, Lucian Chan, Yee Whye Teh, and DS. **Noise Contrastive Meta Learning for Conditional Density Estimation using Kernel Mean Embeddings**. *ArXiv e-prints:1906.02236*, appearing in NeurIPS Meta Learning Workshop 2019.

Meta Learning Setup



Beyond Functional Relationships



- In supervised learning, we often focus on functional relationships, e.g. conditional expectations $\mathbb{E}[y|x]$ in regression.
- More expressive representation may be needed due to e.g. multimodality or heteroscedasticity: y cannot be meaningfully represented using a single function $f(x)$ of the features x, such as $\mathbb{E}[y|x]$.
- Goal: conditional density estimation $p(y|x)$ based on paired samples $\{(x_i, y_i)\}_{i=1}^n$.
- Use a flexible **nonparametric model** of the full conditional density in the **meta learning setting**.

Conditional Mean Embeddings (CME) of Tasks?

- "Augment" the representation of y by using a feature map $\phi_y(y)$ and consider CME $\mathbb{E}[\phi_y(y)|x]$
- We require an expressive feature map ϕ_y so that CME $\mathbb{E}[\phi_y(y)|x]$ captures the relevant information about the relationship between y and x.
- However, CMEs do not give a way to estimate *conditional densities*. Cf. difference between reproducing kernels and smoothing kernels (even in the overlap of the two notions, bandwidths do not go to zero).

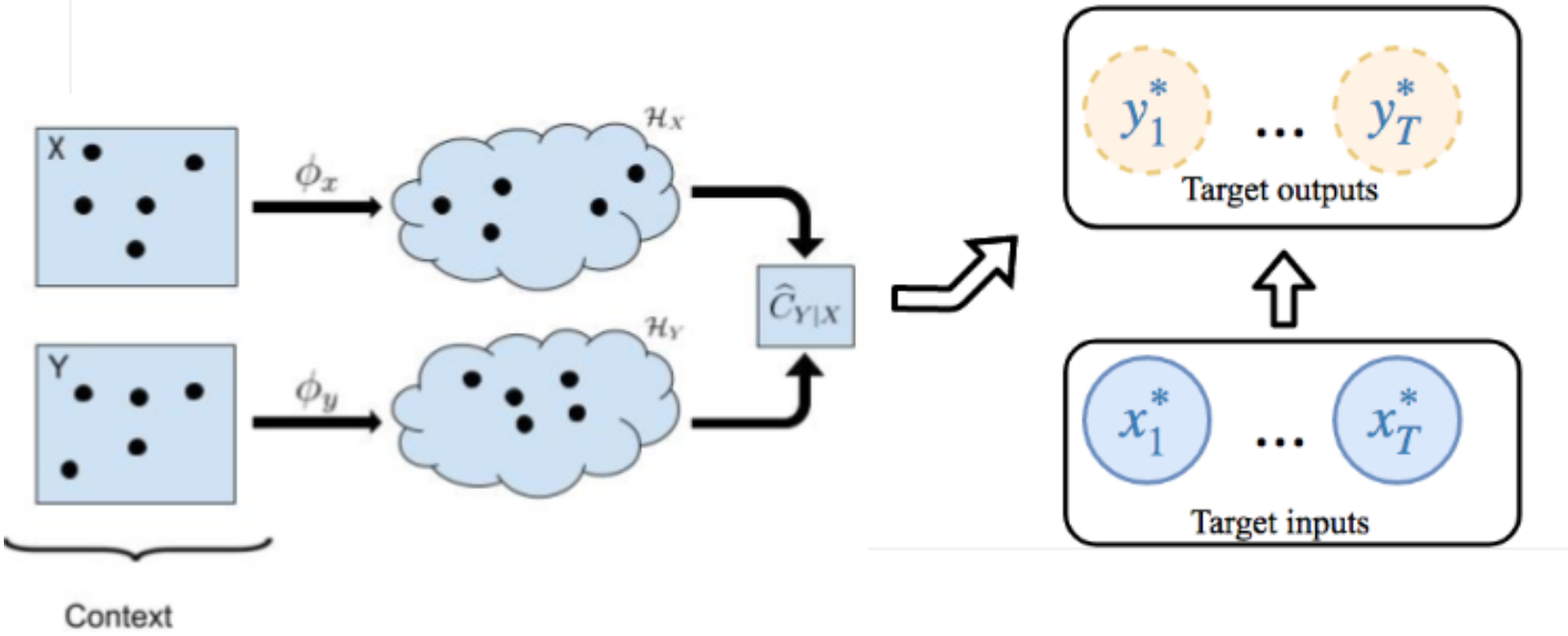
Conditional Mean Embeddings (CME) of Tasks?

- "Augment" the representation of y by using a feature map $\phi_y(y)$ and consider CME $\mathbb{E}[\phi_y(y)|x]$
- We require an expressive feature map ϕ_y so that CME $\mathbb{E}[\phi_y(y)|x]$ captures the relevant information about the relationship between y and x.
- However, CMEs do not give a way to estimate *conditional densities*. Cf. difference between reproducing kernels and smoothing kernels (even in the overlap of the two notions, bandwidths do not go to zero).

- Idea: use the conditional mean embedding operator as a *task embedding* of a given conditional density estimation task.
  - Map x_i and y_i using learned feature maps (neural networks) $\phi_x : \mathcal{X} \to \mathcal{H}_X$ and $\phi_y : \mathcal{Y} \to \mathcal{H}_Y$
  - Estimate the conditional mean embedding operator on the context set:

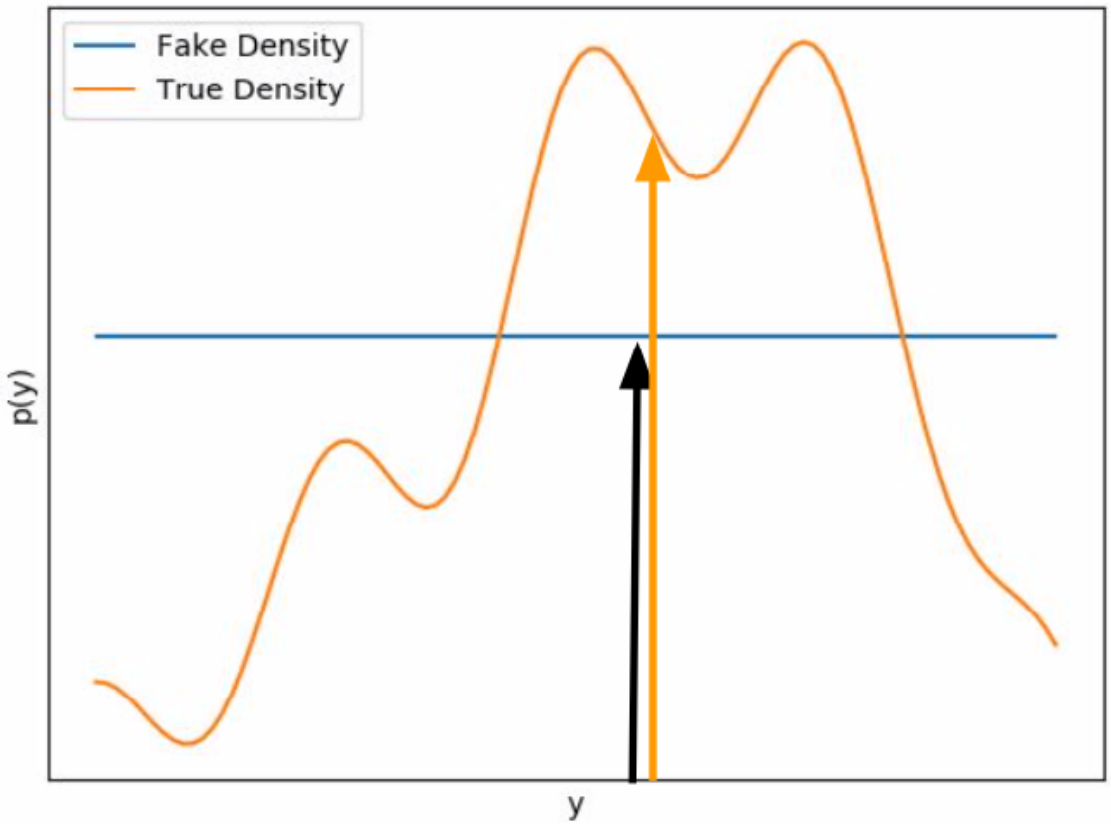
$$\widehat{\mathcal{C}}_{Y|X} = \Phi_y(\underbrace{K_{xx}}_{\Phi_x\Phi_x^\top} + \lambda I)^{-1}\Phi_x^\top, \quad \hat{\mu}_{Y|X=x} = \hat{\mathbb{E}}[\phi_y(y)|x] = \widehat{\mathcal{C}}_{Y|X}\phi_x(x).$$

Proposed Method (so far)



Noise Contrastive Estimation

Noise contrastive estimation [Gutmann and Hyvärinen, 2010] is an approach to the model parameter estimation based on classifiers discriminating between true and artificial (fake) samples.



Noise Contrastive Estimation

Noise contrastive estimation [Gutmann and Hyvärinen, 2010] is an approach to the model parameter estimation based on classifiers discriminating between true and artificial (fake) samples.

In our (conditional) case, let $y_i|x_i \sim p_\theta(y|x)$, and $\{y^f_{i,j}\}_{j=1}^{\kappa} \sim p_f(y)$, for a given "fake" density $p_f(y)$. Giving weights proportional to $(1, \kappa)$, probability that the sample came from the true model is:

$$P_\theta(\text{True}|y, x) = \frac{p_\theta(y|x)}{p_\theta(y|x) + \kappa p_f(y)}.$$

Assuming that the learned classifier is Bayes optimal:

$$p_\theta(y|x) = \frac{\kappa p_f(y) P_\theta(\text{True}|y, x)}{1 - P_\theta(\text{True}|y, x)}.$$

Density model

Consider the density model given by

$$p_\theta(y|x) = \frac{\exp(s_\theta(x,y))}{\int \exp(s_\theta(x,y'))dy'} = \exp(s_\theta(x,y) + b_\theta(x))$$

for some scoring function $s_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and $b_\theta(x)$ models the normalizing constant. Hence

$$\begin{aligned} P_\theta(\text{True}|y,x) &= \frac{\exp(s_\theta(x,y) + b_\theta(x))}{\exp(s_\theta(x,y) + b_\theta(x)) + \kappa p_f(y)} \\ &= \sigma\left(s_\theta(x,y) + b_\theta(x) - \log(\kappa p_f(y))\right). \end{aligned}$$

where $\sigma(t) = 1/(1+e^{-t})$ is the logistic function.

Defining s_θ

- Map x_i and y_i using feature maps (neural networks) $\phi_x : \mathcal{X} \to \mathcal{H}_X$ and $\phi_y : \mathcal{Y} \to \mathcal{H}_Y$ with all parameters collated into θ
- Estimate the conditional mean embedding operator $\widehat{\mathcal{C}}_{Y|X} = \Phi_y (K_{xx} + \lambda I)^{-1} \Phi_x^\top$
- Given $\widehat{\mathcal{C}}_{Y|X}$, we can estimate the conditional mean embedding for any new x' using

$$\hat{\mu}_{Y|X=x'} = \widehat{\mathcal{C}}_{Y|X} \phi_x(x')$$

- We can then evaluate the conditional mean embedding at any new y' using

$$\hat{\mu}_{Y|X=x'}(y') = \langle \hat{\mu}_{Y|X=x'}, \phi_y(y') \rangle_{\mathcal{H}_Y} = \langle \widehat{\mathcal{C}}_{Y|X} \phi_x(x'), \phi_y(y') \rangle_{\mathcal{H}_Y}$$

Scoring function:

$$s_\theta(x', y') = \hat{\mu}_{Y|X=x'}(y')$$

Defining s_θ

- Scoring function:

$$s_\theta(x', y') = \hat{\mu}_{Y|X=x'}(y')$$

- We expect this value to be high when y' is drawn from the true conditional distribution $Y|X = x'$ and low in cases where y' falls in a region where the true conditional density $p(y|x')$ is low:

$$\mu_{Y|X=x'}(y') = \mathbb{E}\left[k_y(y', Y)|X = x'\right] = \int k_y(y', y)p(y|x')dy,$$

where $k_y(y, y') := \langle \phi_y(y), \phi_y(y') \rangle_{\mathcal{H}_y}$.

- Recall that

$$P_\theta(\text{True}|y, x) = \sigma\left(s_\theta(x, y) + b_\theta(x) - \log(\kappa p_f(y))\right).$$

Full density model

In summary, the density model given by

$$p_\theta(y|x) = \frac{\exp(s_\theta(x,y))}{\int \exp(s_\theta(x,y'))dy'} = \exp(s_\theta(x,y) + b_\theta(x))$$

with scoring function

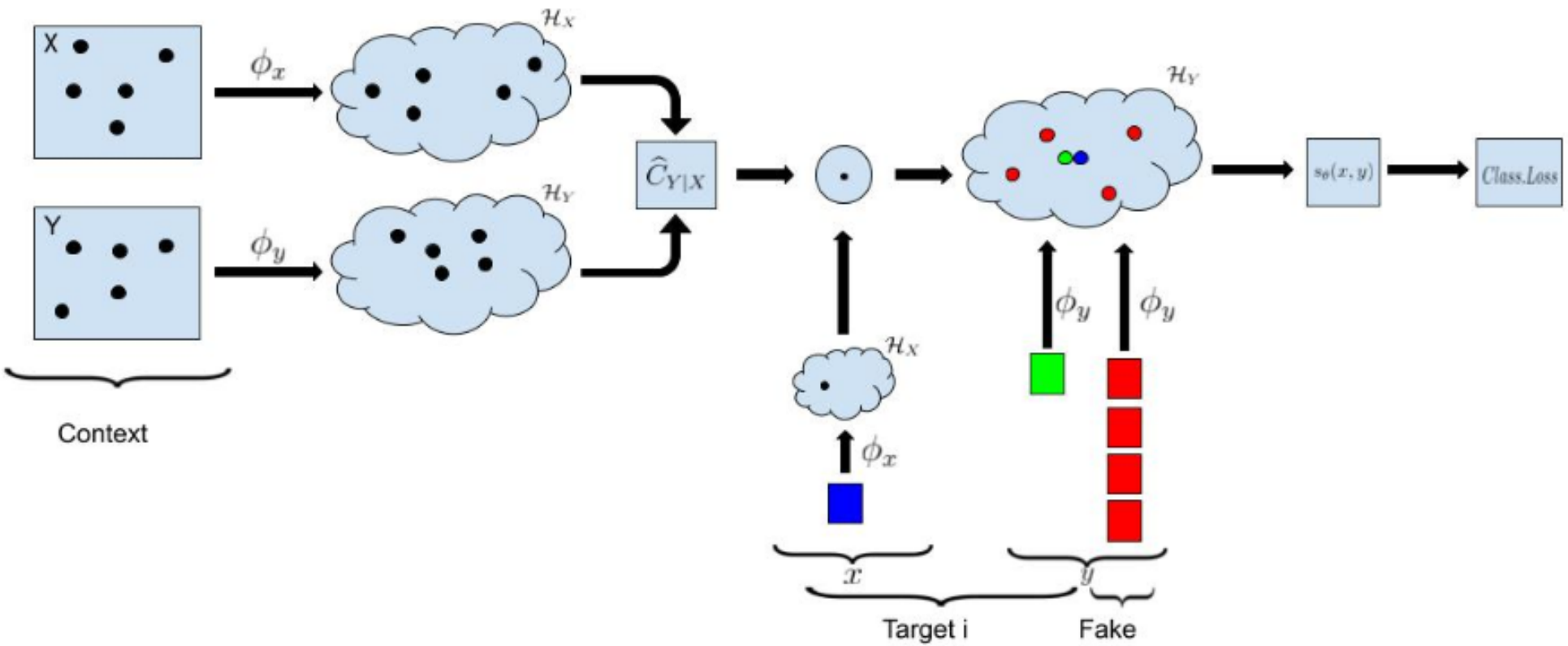
$$s_\theta(x',y') = \hat{\mu}_{Y|X=x'}(y') = \langle \widehat{\mathcal{C}}_{Y|X}\phi_x(x'), \phi_y(y')\rangle_{\mathcal{H}_Y}.$$

and $b_\theta(x)$ is a normalizer network which models the normalizing constant $-\log \int \exp(\hat{\mu}_{Y|X=x}(y'))dy'$,

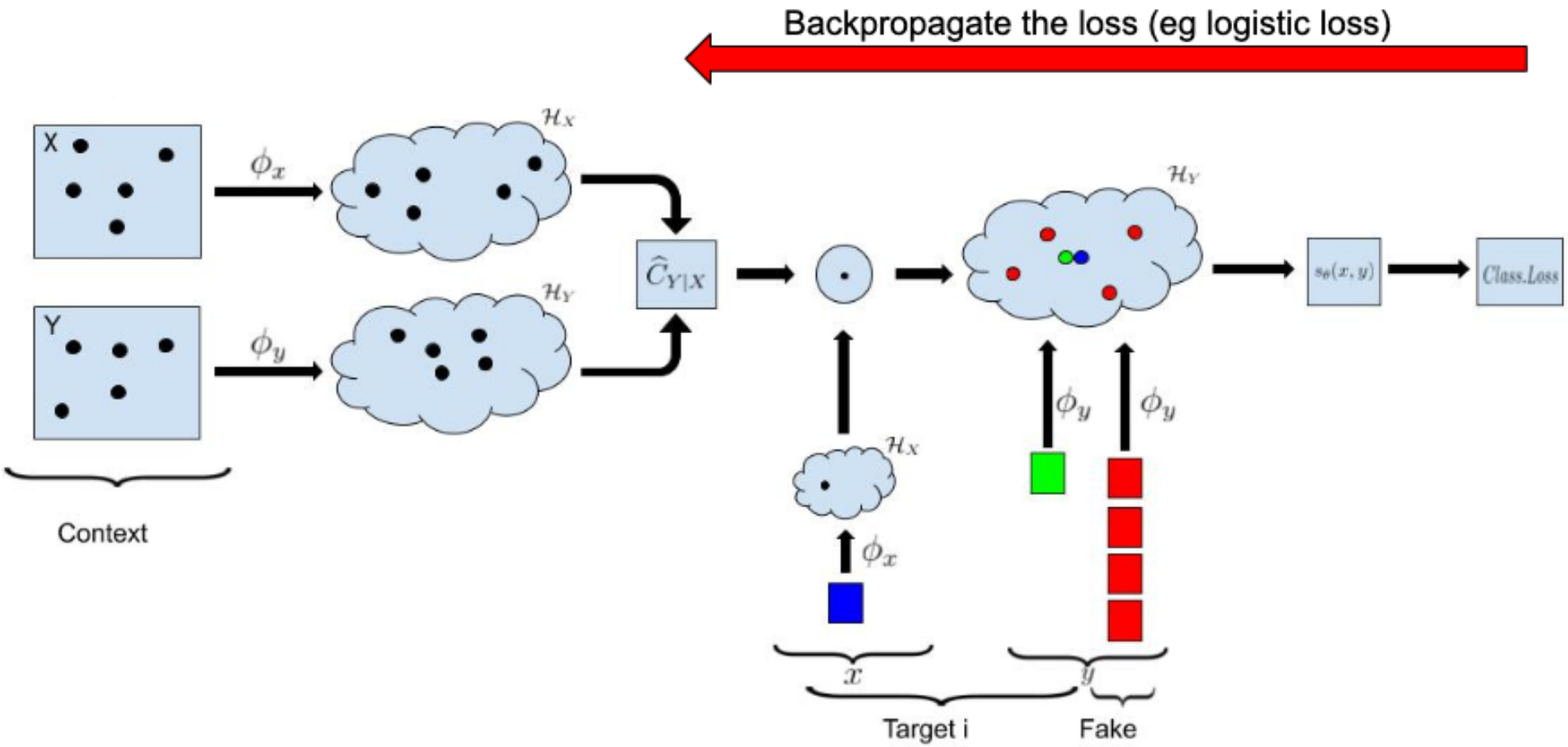
$$\text{so we model } b_\theta(x) = g_\omega(\hat{\mu}_{Y|X=x})$$

with g feedforward network and all parameters (of ϕ_x, ϕ_y and g_ω) collated into θ.

Proposed Method



Proposed Method



MetaCDE

- Three neural networks $\phi_x(x)$, $\phi_y(y)$, $b_\theta(x)$ (all parameters collated into θ) – these will be shared across tasks.
- Let $\mathcal{T} = \{T_1, \ldots, T_L\}$ be the set of L conditional density estimation tasks –each divided into context $\mathcal{D}_c^l = \{(x_i^{l,c}, y_i^{l,c})\}$ and target data $\mathcal{D}_t^l = \{(x_i^{l,t}, y_i^{l,t})\}$
- For every target input $x_i^{l,t}$, we generate κ fake responses $y_{i,j}^{l,f}$ from $p_f(y)$.
- Learn θ by training a True/Fake classifier on the True/Fake labels by minimizing the logistic loss across target data for all tasks jointly (SGD).

$$\min_\theta \sum_l \sum_i \left\{ \log\left(1 + \frac{\kappa p_f(y_i^{l,t})}{\exp(s_\theta^l(x_i^{l,t}, y_i^{l,t}) + b_\theta^l(x_i^{l,t}))}\right) + \sum_{j=1}^{\kappa} \log\left(1 + \frac{\exp(s_\theta^l(x_i^{l,t}, y_{i,j}^{l,f}) + b_\theta^l(x_i^{l,t}))}{\kappa p_f(y_{i,j}^{l,f})}\right) \right\}$$

with

$$s_\theta^l(x, y) = \langle \widehat{\mathcal{C}}_{Y|X}^l \phi_x(x), \phi_y(y) \rangle_{\mathcal{H}_Y}, \quad b_\theta^l(x) = g_\omega(\widehat{\mathcal{C}}_{Y|X}^l \phi_x(x))$$

and $\widehat{\mathcal{C}}_{Y|X}^l$ computed on the context data $\mathcal{D}_c^l$ of the l-th task.

Synthetic data experiment

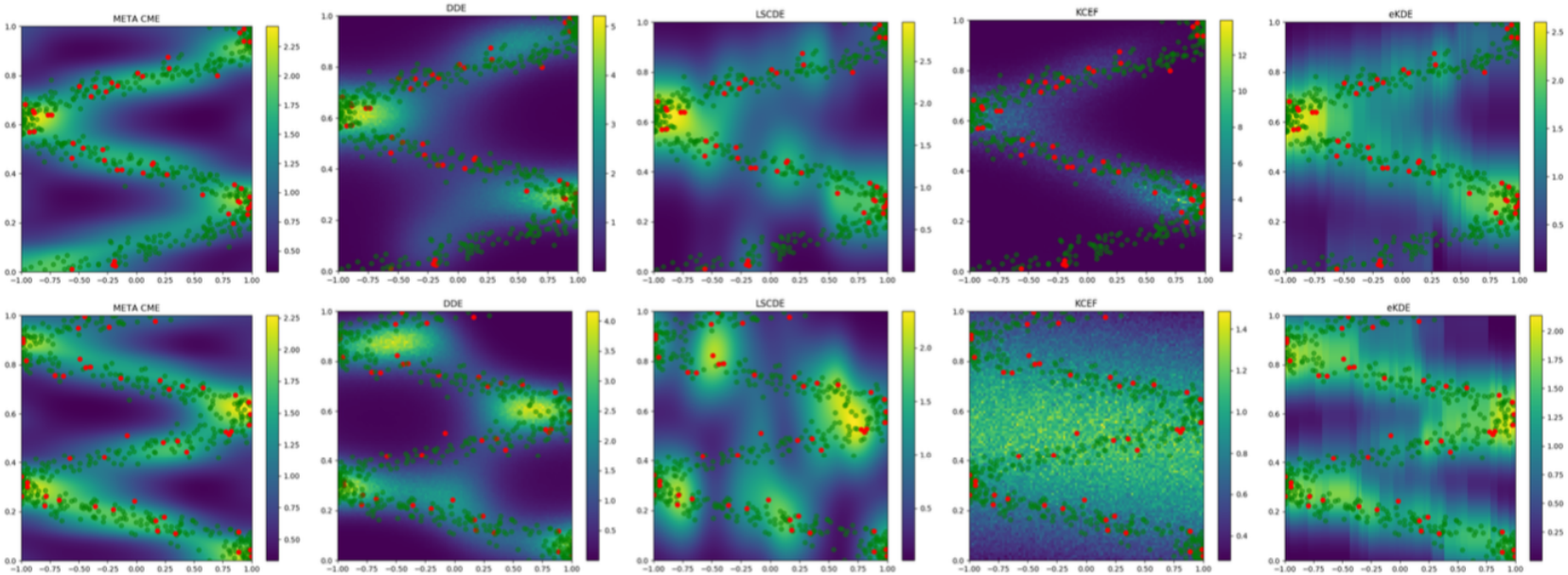


Figure: Left to right: MetaCDE (ours), DDE, LSCDE, KCEF, ϵ-KDE. The red dots are the context/training points and the green dots are points from the true density.

- $y_i \sim \mathrm{U}(0,1)$, $x_i|y_i = \cos(ay_i + b) + \mathcal{N}(0, \sigma^2)$, where a, b, σ^2 vary between tasks.
- Note that in this case x can be written as a simple function of y with added noise, but not vice versa, leading to the multimodality of $p(y|x)$.

Dihedral angles in molecules

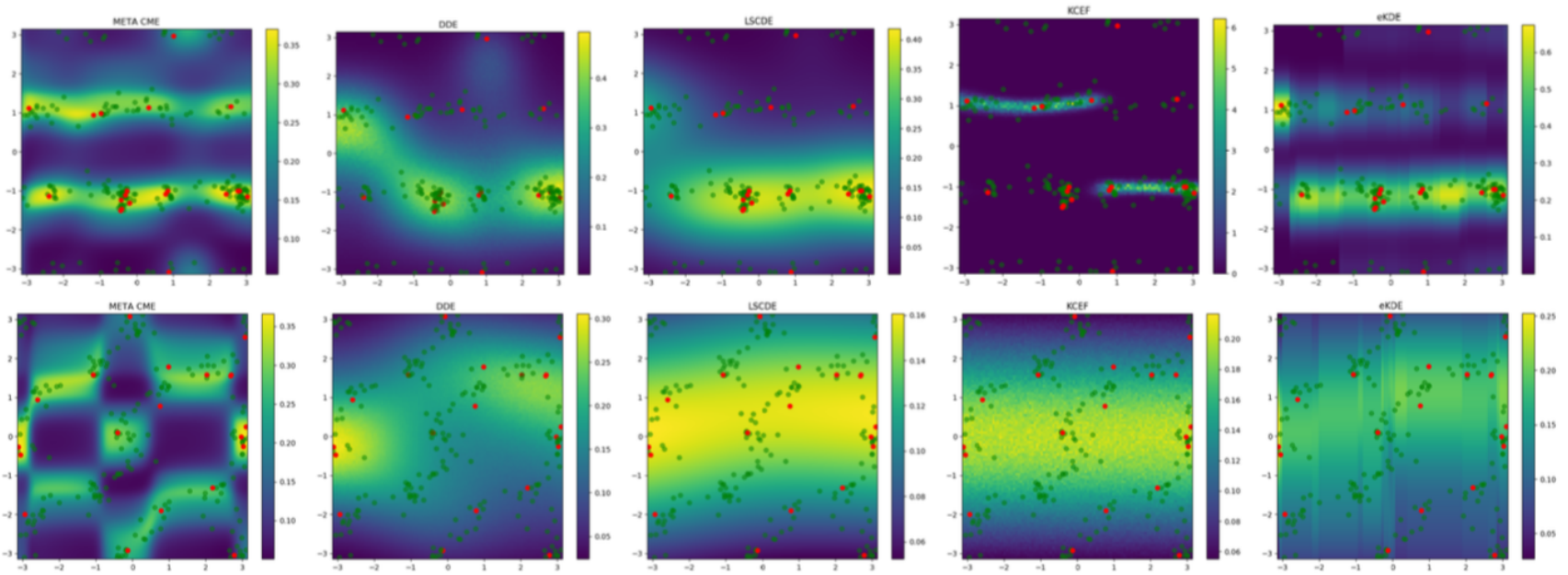


Figure: Left to right: MetaCDE (ours), DDE, LSCDE, KCEF, ϵ-KDE. The red dots are the context/training points and the green dots are points from the true density.

- Interested in understanding possible conformations of molecular structures, i.e. energetically allowed regions of dihedral angles in bonds. The data extracted from crystallography database COD [Gražulis et al, 2011].
- The multimodality of the dataset arises from the molecular symmetries such as reflection and rotational symmetry.

NYC Taxi Dataset

NYC Taxi Dataset (TLC Trip Record Data) from January 2016.
We are interested in estimating $p_{\text{pickup}}(\text{dropoff}|\text{tip})$

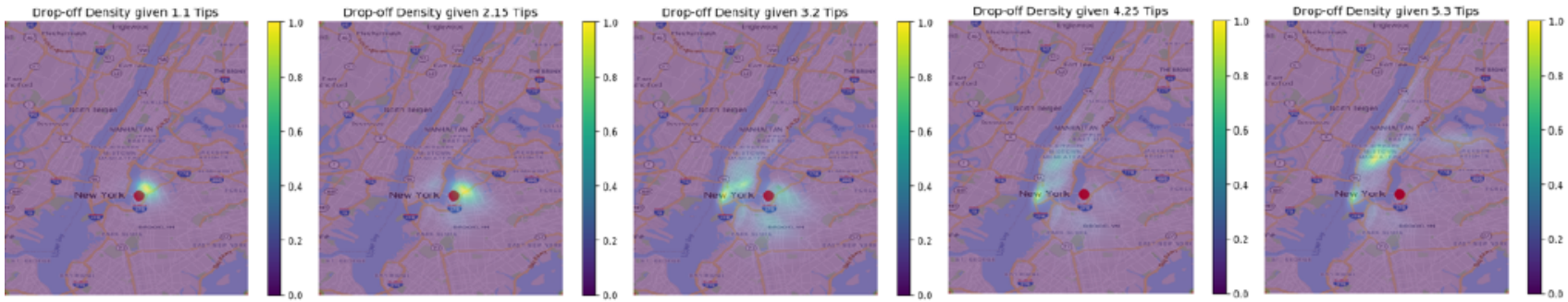


Figure: Red dot: pickup location in Brooklyn not seen by the method (it indexes the task). Estimated density of the dropoff location conditionally on the tip size. As expected, these densities exhibit multimodality.

Data from https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Meta Learning Benchmarks

- **Neural Process**: Used to model regression functions (conditional expectations), so cannot deal with multimodality.
- **MetaNN**: Here, we learn a task embedding using a deep neural network directly applied to the context dataset which is then fed into a score function, with everything else kept the same.

Meta Learning Benchmarks

- **Neural Process**: Used to model regression functions (conditional expectations), so cannot deal with multimodality.
- **MetaNN**: Here, we learn a task embedding using a deep neural network directly applied to the context dataset which is then fed into a score function, with everything else kept the same.

How important is it to encode the structure based on the conditional mean embedding operator?

Results

		Synthetic	Chemistry	NYC Taxi
MetaCDE (Ours)	Loglike	**197.84 ± 22.4**	**-305.49 ± 46.9**	**-1685.52± 608.35**
MetaNN (Ours)	Loglike	132.776±130.87	-317.91±51.3	-2276.55 ± 608.9
	p-value	4.781e-06	1e-03	3.89e-10
Neural Process	Loglike	-81.11±18.5	-426.75± 47.3	-3050.2 ± 822.8
	p-value	<2.2e-16	<2.2e-16	3.89e-10
DDE	Loglike	162.98 ± 69.0	-399.68 ± 41.3	-2236.07 ± 565.9
	p-value	8.14e-07	1.65e-15	3.89e-10
KCEF	Loglike	-388.30 ± 703.1	-724.40 ± 891.6	-1695.89±435.4
	p-value	<2.2e-16	9.72e-14	0.025
LSCDE	Loglike	44.95 ± 74.3	-407.32 ± 80.1	-2748.01 ± 549.2
	p-value	<2.2e-16	2.57e-14	3.89e-10
ϵ-KDE	Loglike	116.31 ± 236.9	-485.10 ± 303.4	-2337.90 ± 501.1
	p-value	2.38e-07	2.94e-14	4.13e-10

Table: Average held out log-likelihood on 100 different tasks. Also reporting the p-values for the one sided signed Wilcoxon test wrt to MetaCDE.

Conclusions

- Learn a data representation informative for the conditional density estimation tasks, by borrowing strength across tasks.
- The approach builds on the probabilistic approaches to meta learning, i.e. neural processes: MetaCDE also learns a task embedding based on the context set, but this embedding takes a specific form of the conditonal embedding operator and it is the feature maps that are learned.
- Combining the feature map networks using kernel mean embedding formalism gives better performance than learning the task embedding directly.

Summary

- Statistical modelling can be brought to bear in tandem with deep learning.
- Increasing confluence between statistics and ML: making use of the well engineered ML infrastructure, with bespoke statistical models for the problem at hand.
- Flexibility of the RKHS framework as a common ground between machine learning and statistical inference.

Kernel Embeddings for Meta Learning

- Jean-Francois Ton, Lucian Chan, Yee Whye Teh, and DS. **Noise Contrastive Meta Learning for Conditional Density Estimation using Kernel Mean Embeddings**. *ArXiv e-prints:1906.02236*, appearing in NeurIPS Meta Learning Workshop 2019.