Mixture of Linear Models(MLM) Co-supervised by Deep Neural Networks

Beomseok Seo

Korea University

Apr. 2022

Introduction

- Statistical models are used everywhere
  - for prediction and empirical verification of variables.

- NOT often used for decision making!
  - because its not accurate and not trustful(interpretable).

Contents

Interpretability - What, Why and How?

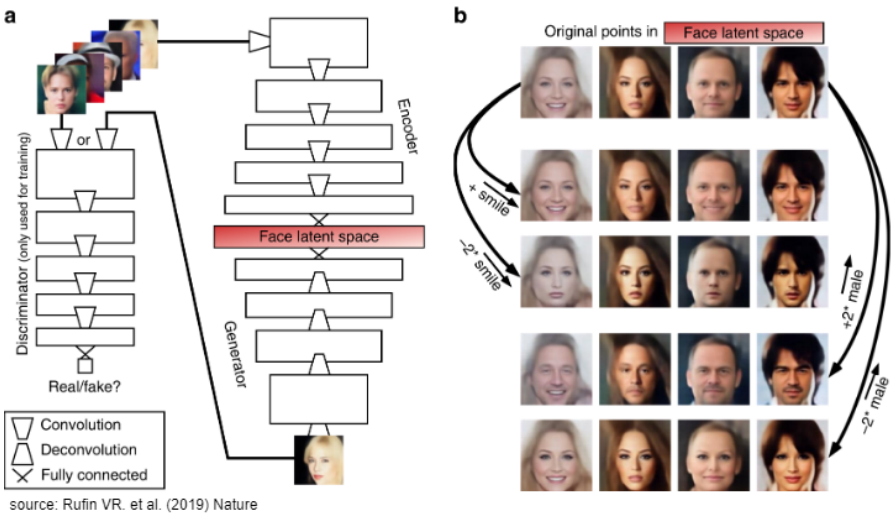
Interpretability - What?

Interpetability: The definition of interpretability is elusive.

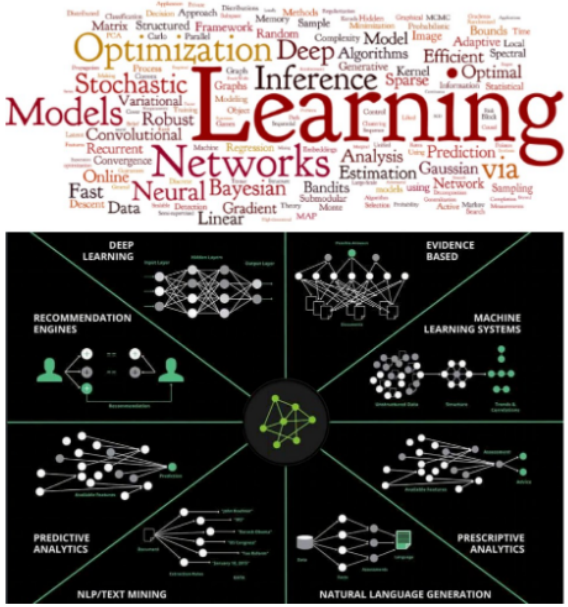
- Interpretability is the degree to which a human can understand the cause of a decision (Miller, 2019)
- Interpretability is the degree to which a human can consistently predict the model's result (Kim et al., 2016)

Interpretability - Why?

- DNN has been successful in many areas.



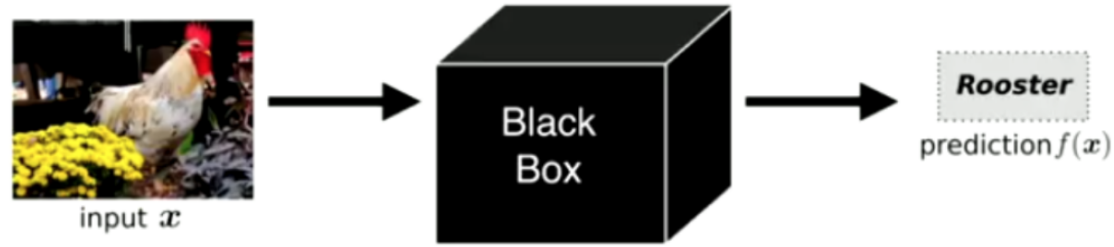
source: Rufin VR. et al. (2019) Nature



source: kadalgo.tech

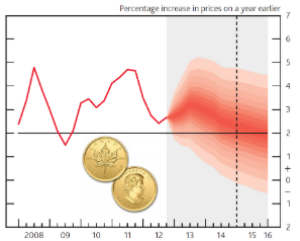
Interpretability - Why?

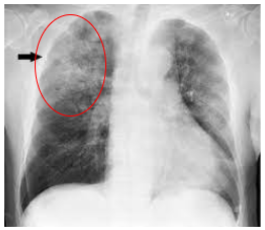
- DNN has met resistance in many areas.



Slide copyright Dr. Wojciech Samek, ODSC Europe 2018

- Interpretation is important if the impact of the results is crucial.



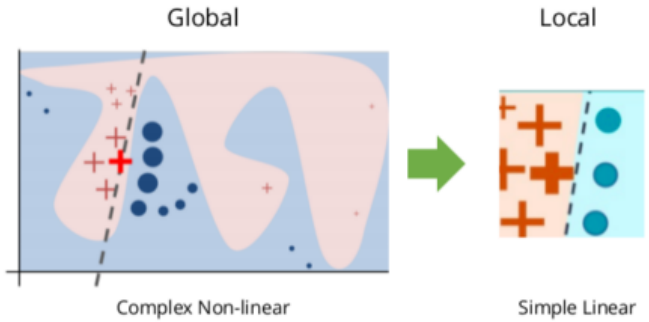


- Models are used not only for the accurate prediction but also for the analysis of variables and the empirical verification of the structure.

Interpretability - How?

- Using post-hoc model-agnostic tools

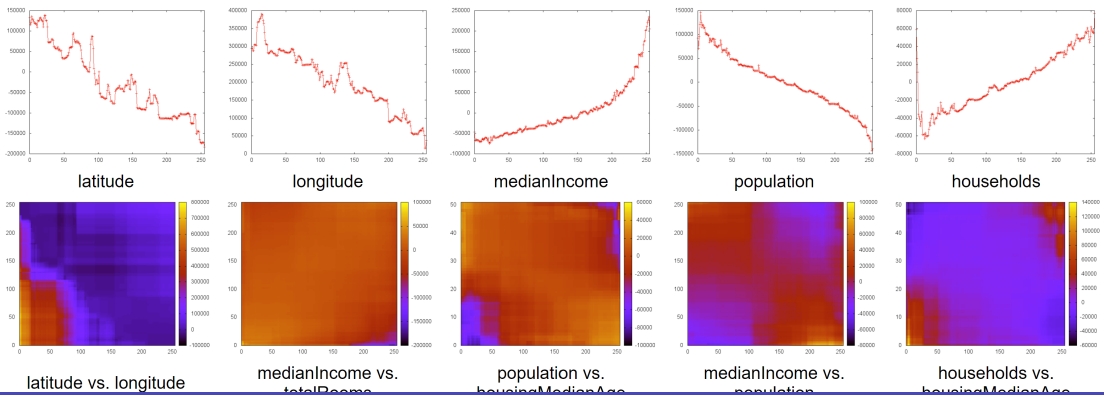
  Locally Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016)



- Building interpretable models by reducing model complexity

  Generalized Additive Models plus Interactions (GA^{2}M)(Lou et al., 2013)

$$g(E[y]) = \sum f_i(x_i) + \sum f_{ij}(x_i, x_j)$$



Mixture of Linear Models (MLM) and Deep Neural Networks(DNN) Overview

Project Overview

- One natural idea to explain a complex model is to view the function as a composite of multiple simple functions in different regions of the input space.

$\Rightarrow$ How to find the segments?

$\Rightarrow$ Using DNN as a proxy of the optimal prediction function.

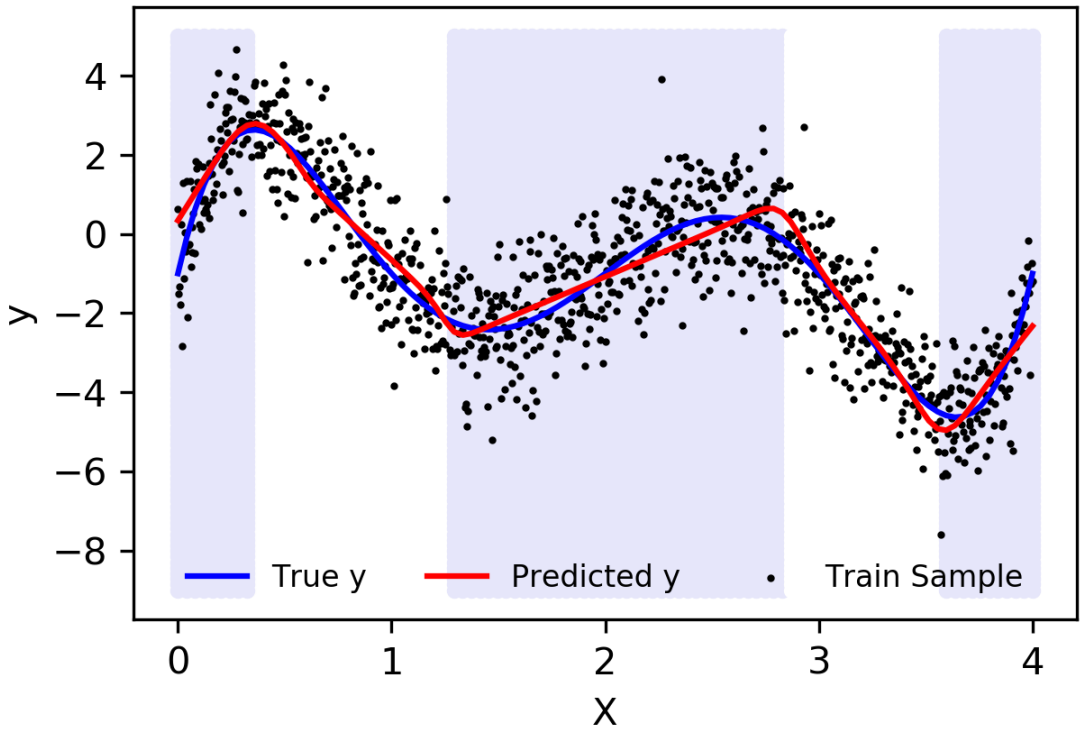


Figure: A toy example of a quintic regression

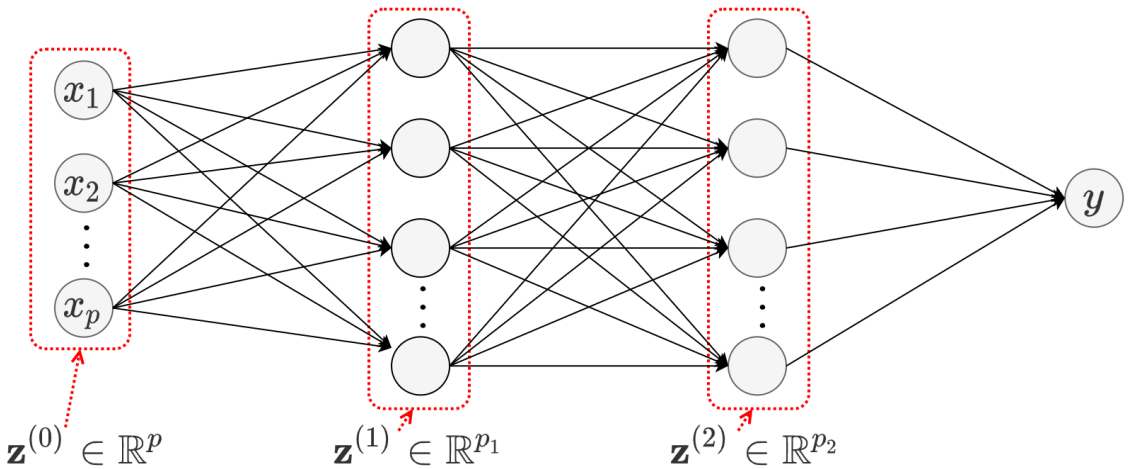
DNN Review

Consider a regression or classification task

- $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}$.
- y_i, $\mathbf{x}_i = (x_{i,1}, \cdots, x_{i,p})^T$, $i = 1, \cdots, n$.

Consider a feed-forward neural network with L hidden layers, $\widetilde{m}(\mathbf{x})$.

- p_l the number of hidden units at l-th layer, $p_0 = p$.
- $\mathbf{z}^{(l)} \in \mathbb{R}^{p_l}$ the outputs of l-th hidden layer, $\mathbf{z}^{(0)} = \mathbf{x}$.



DNN Review

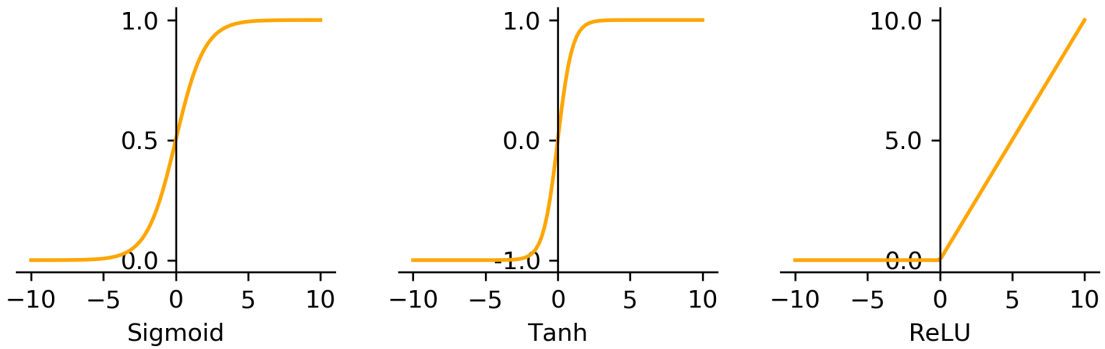
- The mapping at the l-th hidden layer, $\mathbf{z}^{(l)} = h_l(\mathbf{z}^{(l-1)})$, is defined by

$$\begin{aligned} \mathbf{z}^{(l)} &= h_l(\mathbf{z}^{(l-1)}) \\ &= \sigma_l(\mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}), \quad l = 1, \cdots, L, \end{aligned}$$

where $\sigma_l(\cdot)$ is a non-linear element-wise activation function, e.g.,

ReLU function $\max(0, \cdot)$, and

$\mathbf{W}^{(l)} \in \mathbb{R}^{p_l \times p_{l-1}}$ and $\mathbf{b}^{(l)} \in \mathbb{R}^{p_l}$ the model parameters.



DNN Review

- The neural network model, $\widetilde{m}(\mathbf{x})$, is given by

$$\begin{aligned}\widetilde{m}(\mathbf{x}) &= (g \circ h_L \circ h_{L-1} \circ \cdots \circ h_1)(\mathbf{x}) \\ &= g(\sigma_L(\cdots \sigma_2(\mathbf{W}^{(2)}\sigma_l(\mathbf{W}^{(l)}\mathbf{x} + \mathbf{b}^{(l)}) + \mathbf{b}^{(2)})\cdots)).\end{aligned}$$

- Universal approximation theorems imply that neural networks can represent a wide variety of functions.

DNN Review

- When ReLU (rectified linear unit) is used for the activation functions, a DNN is piecewise linear.

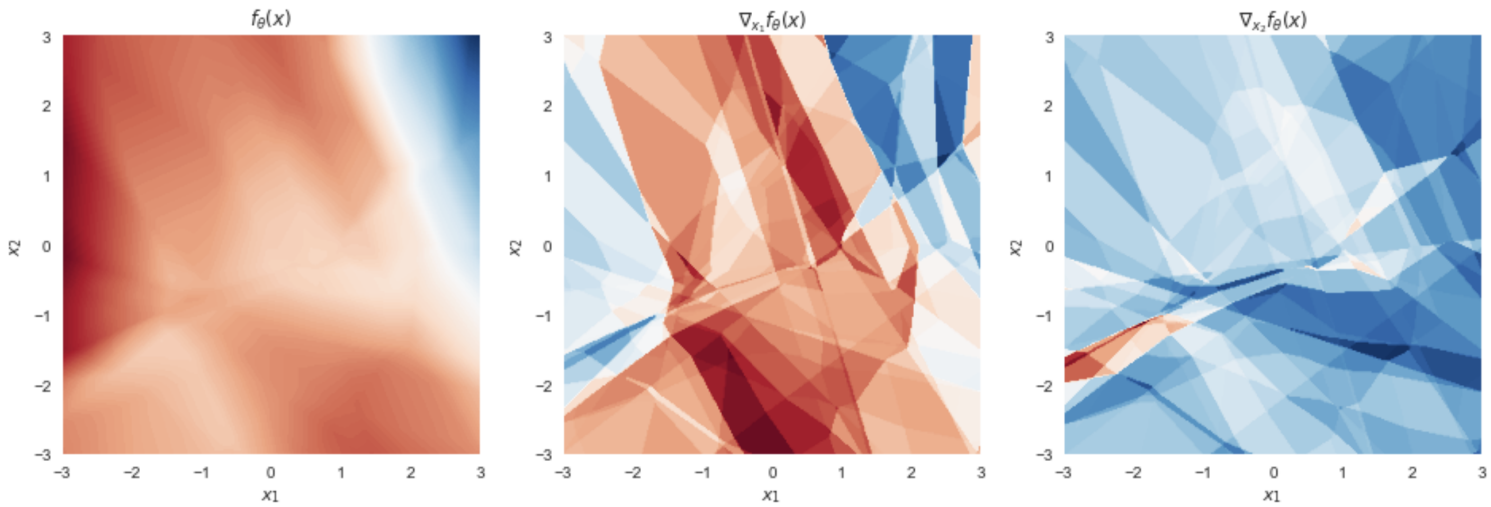


Figure: For a DNN model $\widetilde{m}$, (left) predicted values $\widetilde{m}(x_1, x_2)$, (center) partial derivative w.r.t. x_1, $\nabla_{x_1}\widetilde{m}(x_1, x_2)$, (right) partial derivative w.r.t. x_2, $\nabla_{x_2}\widetilde{m}(x_1, x_2)$

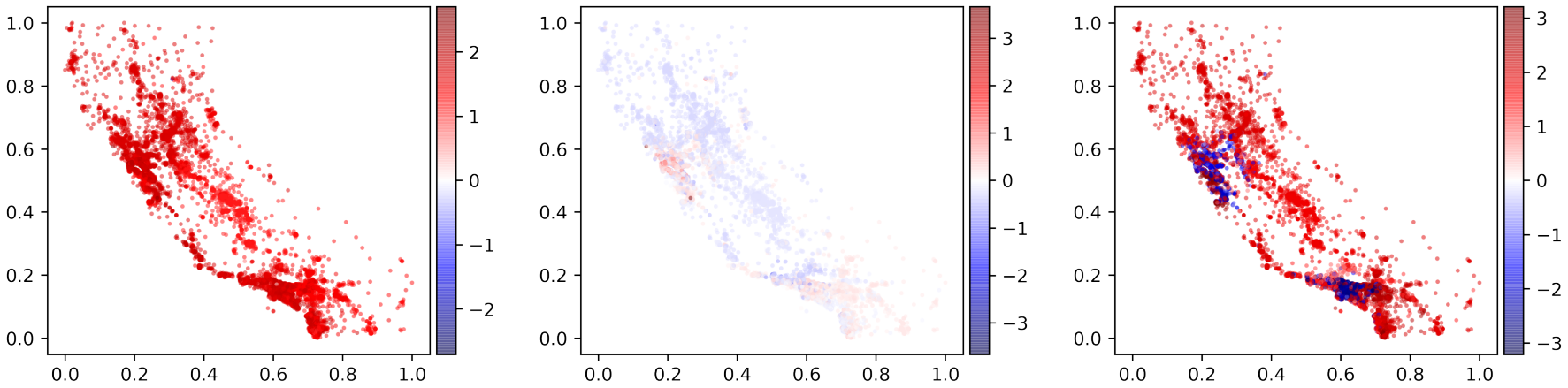
Mixture of Linear Models (MLM)

- *Mixture of Linear Models (MLM)*.
  - $\mathcal{X}$ is divided into $\widetilde{J}$ MECE sets, $\mathscr{P} = \{\mathcal{P}_1, ..., \mathcal{P}_{\widetilde{J}}\}$.

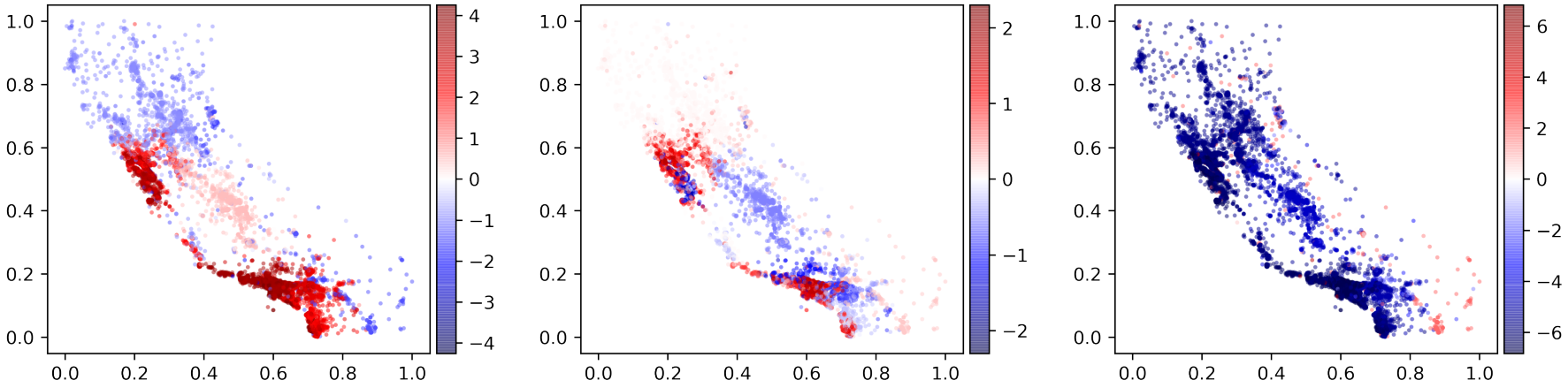
$$\begin{aligned} \hat{m}(\mathbf{x}) &= \gamma_1(\mathbf{x})m_1(\mathbf{x}) + \cdots + \gamma_{\widetilde{J}}(\mathbf{x})m_{\widetilde{J}}(\mathbf{x}), \\ m_j(\mathbf{x}) &= \alpha_j + \mathbf{x}^T\beta_j, \qquad j = 1, ..., \widetilde{J}. \end{aligned} \tag{1}$$

where $m_j(\mathbf{x})$ is *local linear model* approximating $\widetilde{m}(\mathbf{x})$ within $\mathcal{P}_j$,
$\gamma_j(\mathbf{x}) = P(X \in \mathcal{P}_j | X = \mathbf{x})$ is a weight for $m_j(\mathbf{x})$.

MLM: Cal Housing - Regression Coefficients



(a) Median income (b) House age (c) Other rooms



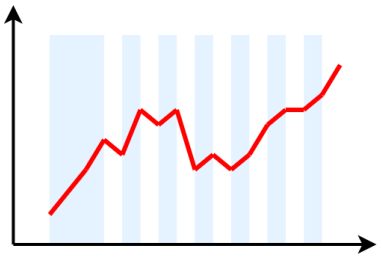
(d) Bedrooms (e) Population (f) Occupancy

Figure: Log-valued regression coefficients of local linear models plotted on longitude (horizontal axis) and latitude (vertical axis) space.

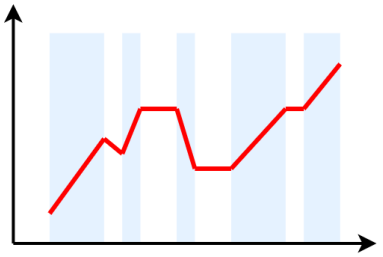
Construction and Interpretation of MLM

Mixture of Linear Models (MLM)

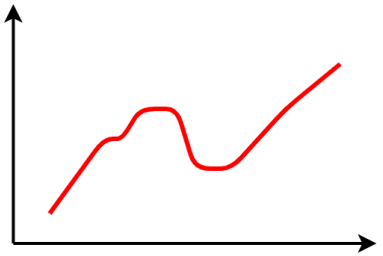
Step 1: Approximate DNN by a piecewise linear function



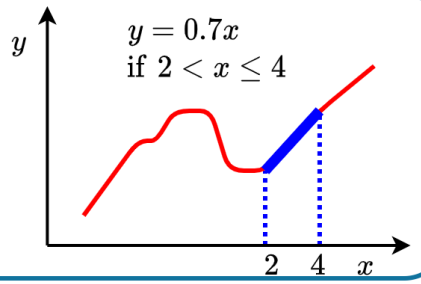
Step 2: Merge the clusters into the bigger clusters



Step 3: Compute soft-weights for each piecewise linear function

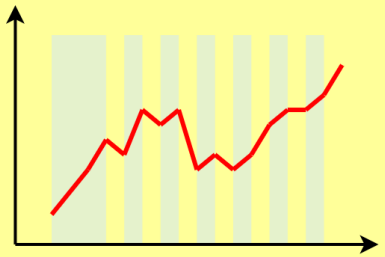


Interpretation: Visualize and describe clusters

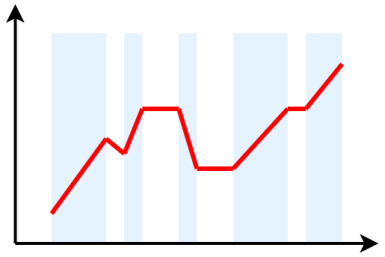


Mixture of Linear Models (MLM)

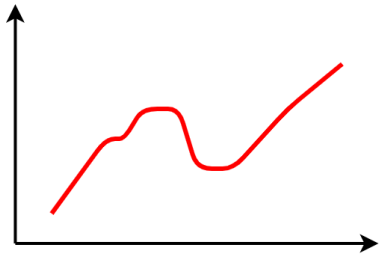
Step 1: Approximate DNN by a piecewise linear function



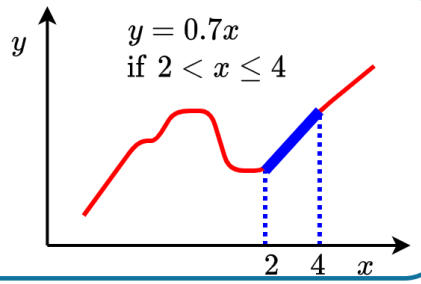
Step 2: Merge the clusters into the bigger clusters



Step 3: Compute soft-weights for each piecewise linear function

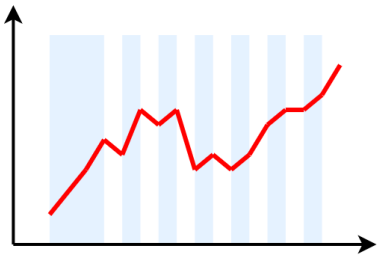


Interpretation: Visualize and describe clusters

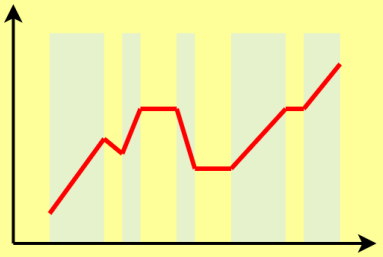


Mixture of Linear Models (MLM)

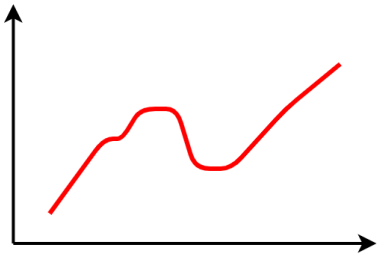
Step 1: Approximate DNN by a piecewise linear function



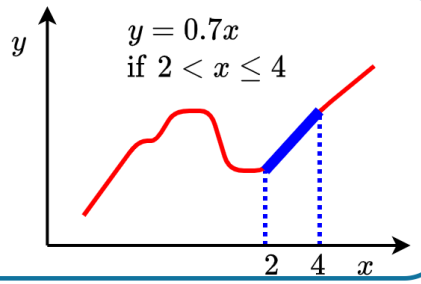
Step 2: Merge the clusters into the bigger clusters



Step 3: Compute soft-weights for each piecewise linear function

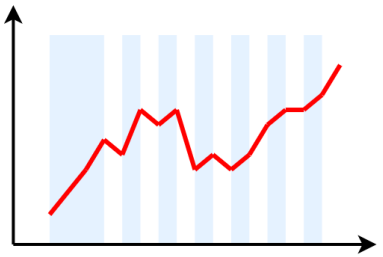


Interpretation: Visualize and describe clusters

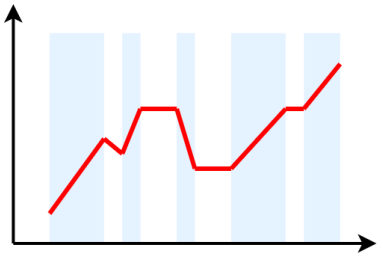


Mixture of Linear Models (MLM)

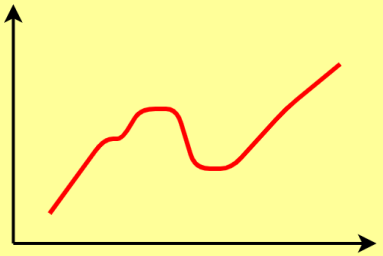
Step 1: Approximate DNN by a piecewise linear function



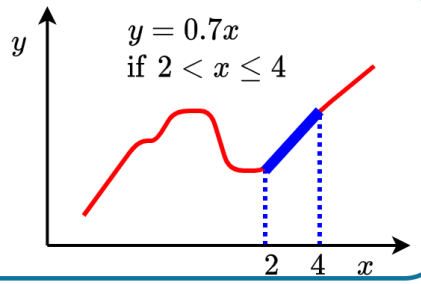
Step 2: Merge the clusters into the bigger clusters



Step 3: Compute soft-weights for each piecewise linear function

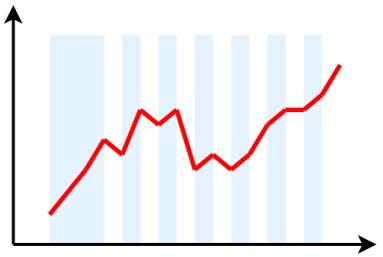


Interpretation: Visualize and describe clusters

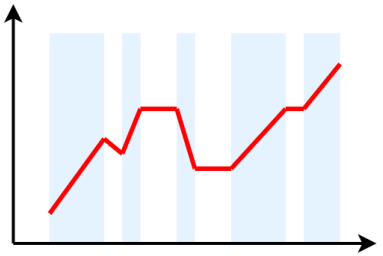


Mixture of Linear Models (MLM)

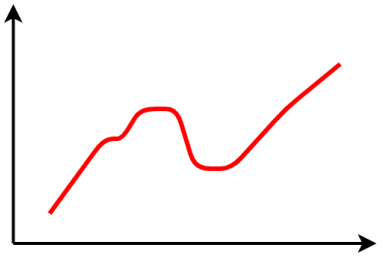
Step 1: Approximate DNN by a piecewise linear function



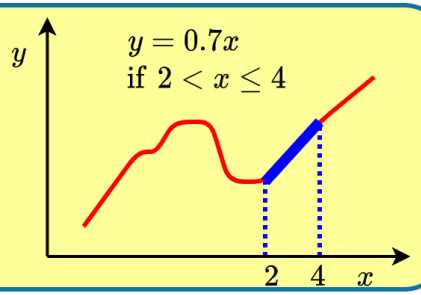
Step 2: Merge the clusters into the bigger clusters



Step 3: Compute soft-weights for each piecewise linear function



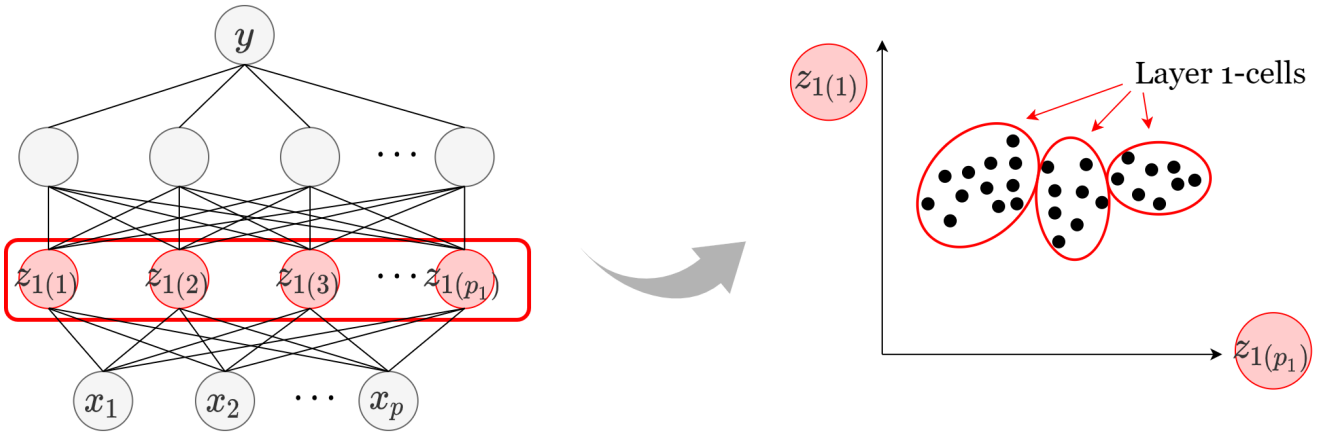
Interpretation: Visualize and describe clusters



Step 1: Piecewise Linear Approximation of DNN

$\Rightarrow$ Cluster the outputs $\mathbf{z}^{(l)}$ at each layer of the DNN into K_l clusters, $\{\mathcal{C}_1^{(l)}, \cdots, \mathcal{C}_{K_l}^{(l)}\}$, called *layer l-cells*.

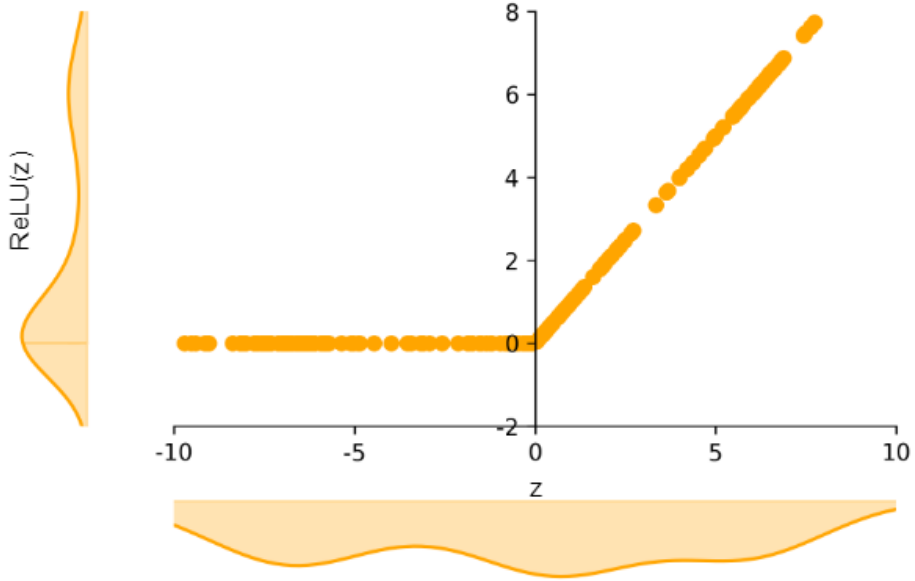
$$f(\mathbf{z}^{(l)}) = \sum_{k=1}^{K_l} \pi_k^{(l)} \phi(\mathbf{z}^{(l)} | \mu_k^{(l)}, \Sigma_k^{(l)}).$$



Step 1: Piecewise Linear Approximation of DNN

- Note that each hidden layer, $h_l(\mathbf{z}^{(\mathbf{l-1})})$, is monotone and approximately piecewise linear:

$$\begin{aligned}\hat{h}_l(\mathbf{z}^{(l-1)}) &= I_{\mathcal{C}_1^{(l)}}(\mathbf{z}^{(l-1)})(\mathbf{W}_1^{(1)}\mathbf{z}^{(l-1)} + \mathbf{b}_1^{(1)}) + \\ &\quad \cdots + I_{\mathcal{C}_{K_l}^{(l)}}(\mathbf{z}^{(l-1)})(\mathbf{W}_{K_l}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}_{K_l}^{(l)}),\end{aligned}$$



Step 1: Piecewise Linear Approximation of DNN

$\Rightarrow$ Connect the layer l-cells across the L layers by Cartesian product.

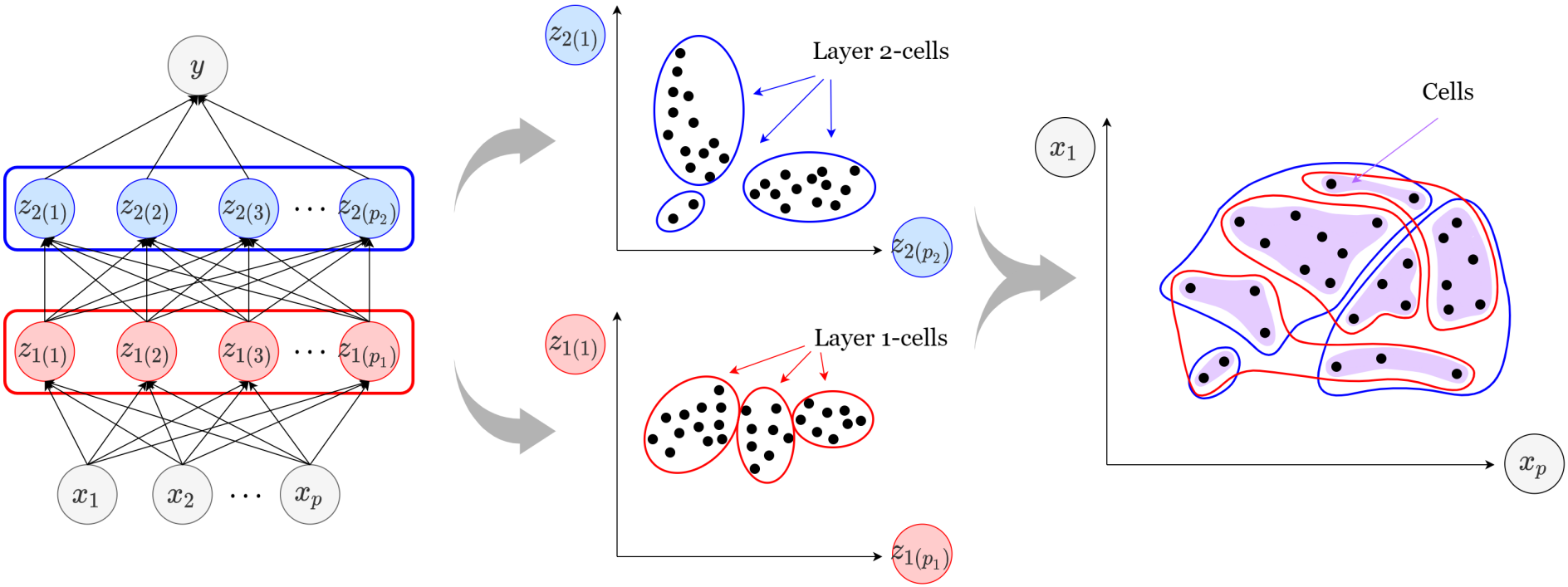
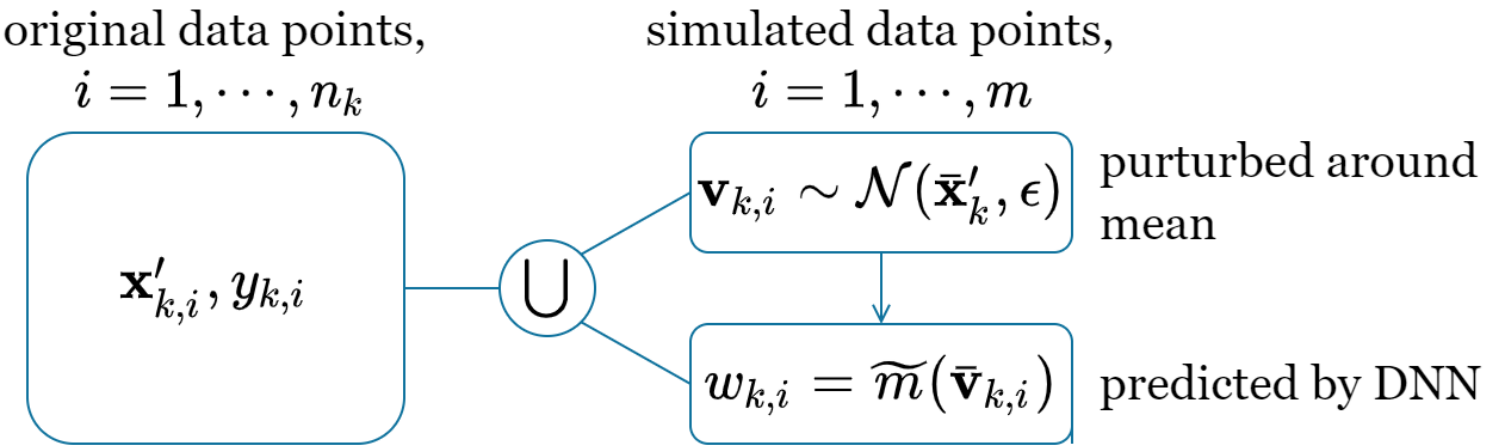


Figure: Forming cells from a neural network

Step 1: Piecewise Linear Approximation of DNN

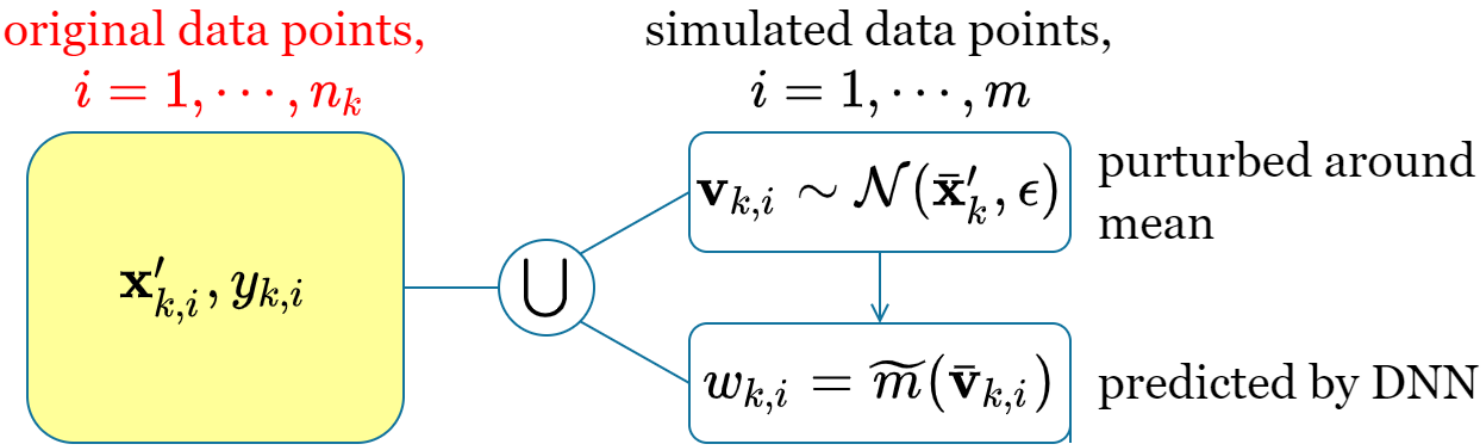
- To train each local linear model $m_k(\mathbf{x})$, we use



$$\Rightarrow \qquad w'_{k,i} = \hat{\alpha}_k + {\mathbf{v}'_{k,i}}^T \hat{\beta}_k + \epsilon_{k,i}$$

Step 1: Piecewise Linear Approximation of DNN

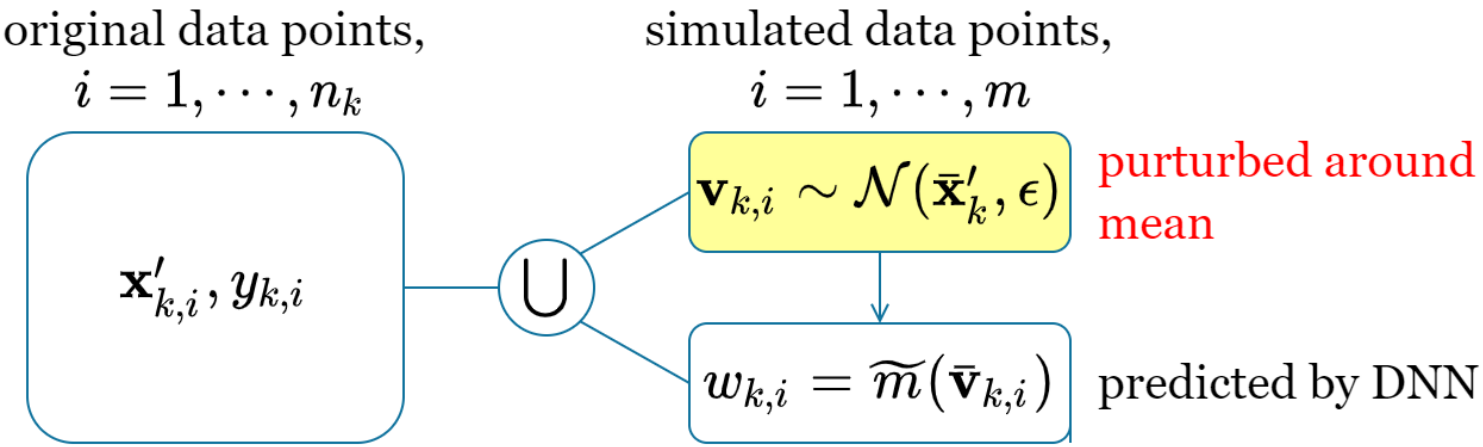
- To train each local linear model $m_k(\mathbf{x})$, we use



$$\Rightarrow \qquad w'_{k,i} = \hat{\alpha}_k + {\mathbf{v}'_{k,i}}^T \hat{\beta}_k + \epsilon_{k,i}$$

Step 1: Piecewise Linear Approximation of DNN

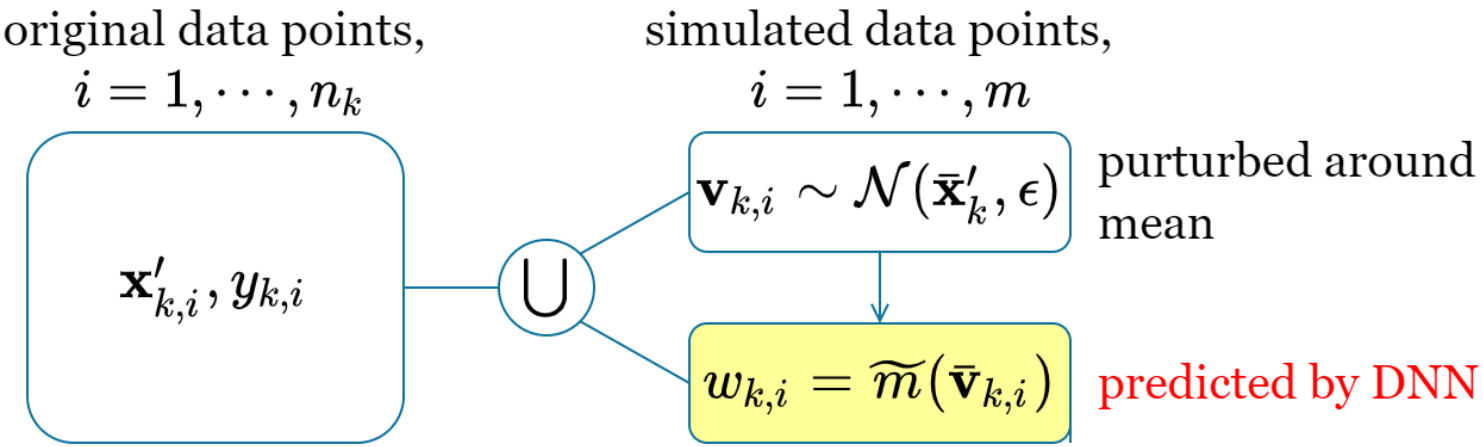
- To train each local linear model $m_k(\mathbf{x})$, we use



$$\Rightarrow \qquad w'_{k,i} = \hat{\alpha}_k + {\mathbf{v}'_{k,i}}^T \hat{\beta}_k + \epsilon_{k,i}$$

Step 1: Piecewise Linear Approximation of DNN

- To train each local linear model $m_k(\mathbf{x})$, we use



$$\Rightarrow \qquad w'_{k,i} = \hat{\alpha}_k + \mathbf{v}'^{T}_{k,i} \hat{\beta}_k + \epsilon_{k,i}$$

Step 1: Piecewise Linear Approximation of DNN

- Approximate DNN($\widetilde{m}(\mathbf{x})$) by a piecewise linear function.

$$\begin{aligned} \hat{m}'(\mathbf{x}) &= I_{\mathcal{C}_1}(\mathbf{x})m_1(\mathbf{x}) + \cdots + I_{\mathcal{C}_{\widetilde{K}}}(\mathbf{x})m_{\widetilde{K}}(\mathbf{x}), \\ m_k(\mathbf{x}) &= \alpha_k + \mathbf{x}^T\beta_k, \quad \text{for } k = 1, \cdots, \widetilde{K}. \end{aligned}$$

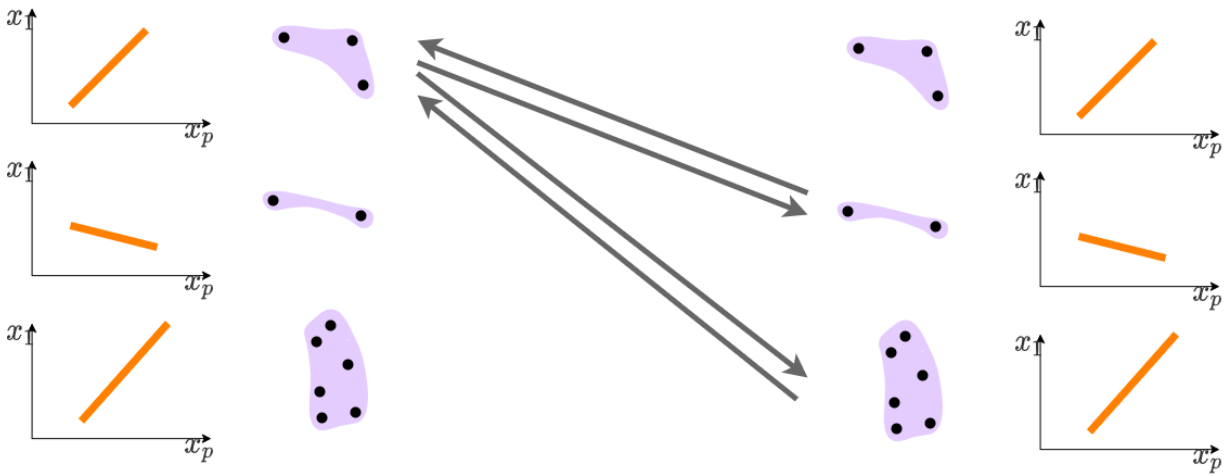
where $\beta_k \in \mathbb{R}^p$ and $\alpha_k \in \mathbb{R}$ are model parameters,

$I_{\mathcal{C}_k}(\mathbf{x})$ is the indicator function.

Step 2: MLM based on EPICs

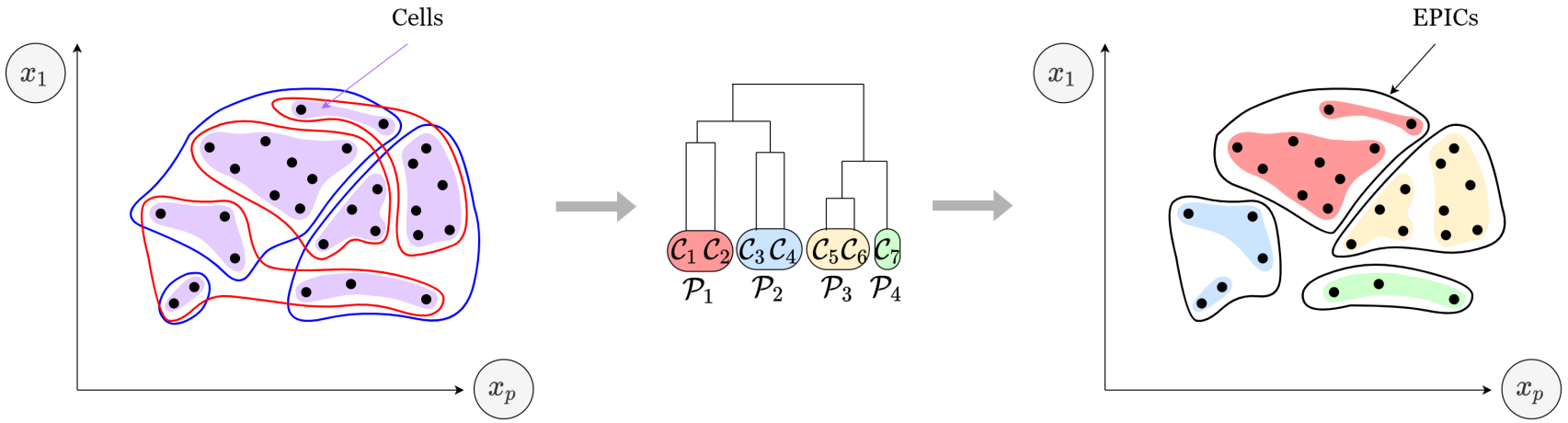
- Define *mutual prediction disparity*, $d_{s,t}$, between the pair of local linear models $m_s(\mathbf{x})$ and $m_t(\mathbf{x})$, $s, t \in \{1, ..., \widetilde{K}\}$ by

$$d_{s,t} = \frac{1}{n_s + n_t + 2m} \left[\sum_{i=1}^{n_s+m} \left(m_s(\mathbf{v}'_{s,i}) - m_t(\mathbf{v}'_{s,i}) \right)^2 + \sum_{i=1}^{n_t+m} \left(m_s(\mathbf{v}'_{t,i}) - m_t(\mathbf{v}'_{t,i}) \right)^2 \right].$$



Step 2: MLM based on EPICs

- Merge the cells($\mathcal{C}_k$) into a user-specified $\widetilde{J}$ clusters using hierarchical clustering with $d_{s,t}$ as the distance.



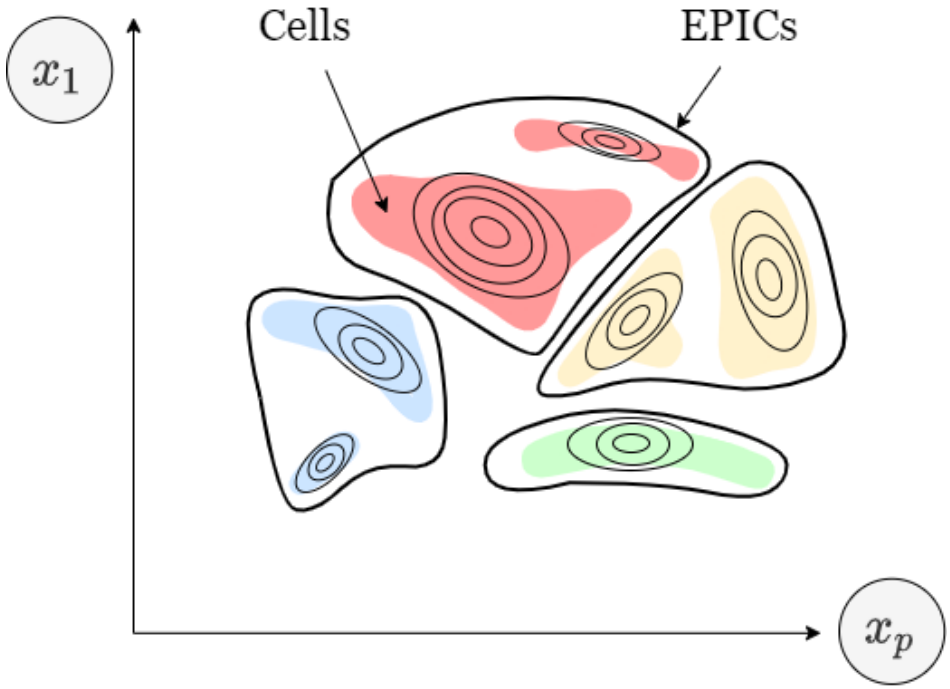
- Define *EPIC (Explainable Prediction-induced Input Cluster)*. For $j = 1, ..., \widetilde{J}$,

$$\mathcal{P}_j = \bigcup_{k \in \mathcal{J}_j} \mathcal{C}_k \,.$$

Step 3: Soft-weighted MLM based on EPIC

$\Rightarrow$ Smoothed function has a lower error rate.

- Fit a single Gaussian densities for each cell $\mathcal{C}_k$,
  combine them as a GMM for each EPIC $\mathcal{P}_j$ to use as the soft-weights



Step 3: Soft-weighted MLM based on EPIC

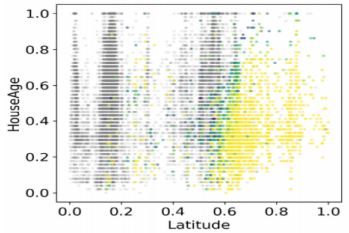
- Then, the final soft-weighted MLM becomes

$$\begin{aligned} \hat{m}(\mathbf{x}) &= \sum_{j=1}^{\widetilde{J}} \gamma_j(\mathbf{x}) m_j(\mathbf{x}) \\ m_j(\mathbf{x}) &= \alpha_j + \mathbf{x}^T \beta_j, \quad j = 1, ..., \widetilde{J}, \\ \gamma_j(\mathbf{x}) &= \frac{\tilde{\pi}_j f_{\mathcal{P}_j}(\mathbf{x})}{\sum_{j'=1}^{\widetilde{J}} \tilde{\pi}_{j'} f_{\mathcal{P}_{j'}}(\mathbf{x})}. \end{aligned} \tag{2}$$

Interpretation of MLM

Interpretation

Low Dimensional Subspace (LDS) :
Visualization of EPICs

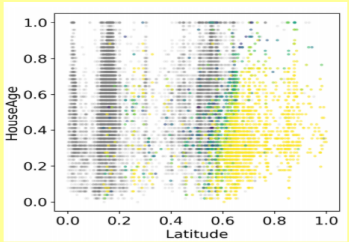


Prominent Region (PR) :
Descriptive rules of EPICs

Samples in EPIC 1 are
- raining = 1
- temperature = (45, 60)
- season = spring

Interpretation

Low Dimensional Subspace (LDS) :
Visualization of EPICs

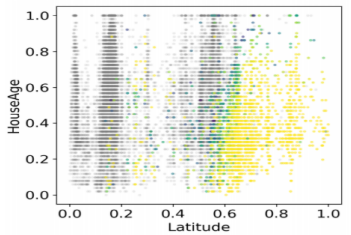


Prominent Region (PR) :
Descriptive rules of EPICs

Samples in EPIC 1 are
- raining = 1
- temperature = (45, 60)
- season = spring

Interpretation

Low Dimensional Subspace (LDS) :
Visualization of EPICs



Prominent Region (PR) :
Descriptive rules of EPICs

Samples in EPIC 1 are
- raining = 1
- temperature = (45, 60)
- season = spring

Interpretation: Low Dimensional Subspace (LDS)

- Denote by $\mathbf{s}$: a subset of variable indices, $\mathbf{s} \subseteq \{1, ..., p\}$,
  $X_{[\mathbf{s}]}$: the subvector of X specified by $\mathbf{s}$.

- The joint marginal density of EPIC $\mathcal{P}_j$ based on $X_{[\mathbf{s}]}$ is Gaussian mixture.

$$\tilde{\pi}_j f_{\mathcal{P}_j,\mathbf{s}}(\mathbf{x}_{[\mathbf{s}]})$$

- Compute two binary indicator labels for EPIC $\mathcal{P}_j$ by

$\mathbf{q}_j$ based on the entire space X
$\hat{\mathbf{q}}_j$ based on the subspace $X_{[\mathbf{s}]}$

Interpretation: Low Dimensional Subspace (LDS)

- Search LDS, $\mathbf{s}_j^*$, for EPIC $\mathcal{P}_j$ in a greedy manner.

$$\begin{aligned} \mathbf{s}_j^* \quad &= \quad \arg\min_{\mathbf{s}} |\mathbf{s}| \\ &\text{s.t.} \quad F_1(\mathbf{q}_j, \hat{\mathbf{q}}_j) \geq \xi \\ &\text{where} \quad \xi \text{ is a given threshold.} \end{aligned}$$

- We call $\mathbf{s}_j^*$ as explainable dimension and $F_1(\mathbf{q}_j, \hat{\mathbf{q}}_j)$ at $\mathbf{s}_j^*$ as explainable rate

Interpretation: Low Dimensional Subspace (LDS)

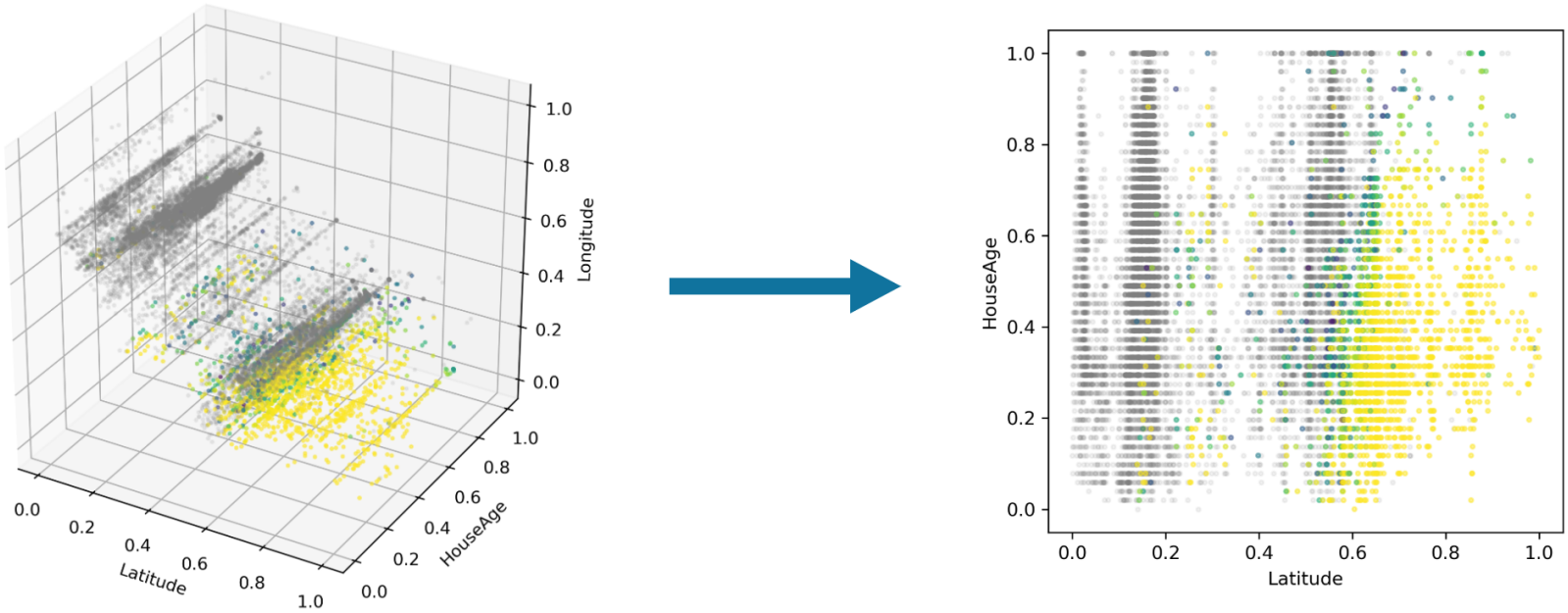


Figure: Low Dimensional Subspace (LDS)

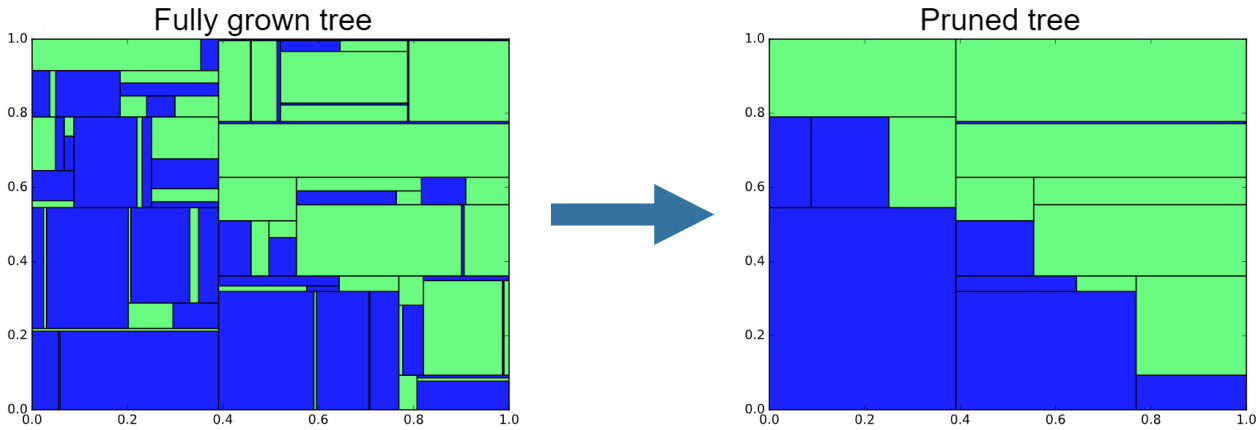
Interpretation: Prominent Region (PR)

- Fit a fully grown decision tree $\mathcal{D}_j$ for EPIC $\mathcal{P}_j$ based on $\{\mathbf{x}_i, q_{j,i}\}_{i=1}^n$.

- Prune $\mathcal{D}_j$ as long as the leaf nodes have at least higher than a given purity threshold ψ:

$$\frac{\sum_{i=1}^{n} I_{u(e)}(i) q_{j,i}}{s(e)} \geq \psi \,,$$

where e is a node in $\mathcal{D}_j$ and $s(e)$ is the *size* of e.



Interpretation: Prominent Region (PR)

- Collect all the leaf nodes $\hat{e}_\tau$ in $\mathcal{D}_j$ satisfying
  (1) its purity being above ψ
  (2) its size being above a pre-chosen minimum size threshold η.

- The decision paths of $\hat{e}_\tau$'s provide descriptive rules for the EPIC $\mathcal{P}_j$. We call them as *explainable conditions*.

Experiment Results of MLM

Experiments

Data	TCGA SKCM	Parkinson's Disease	Bike Sharing	Cal. Housing
Task	classification		regression	
Type	clinical		time series	spatial
Target variable	overall survival status	patients or not	rental counts	house values
Num. of samples	388	756	17379	20640
Num. of orig. features	30	753	12	8
After dummy encoding	73	753	16	8
Num. of layer l-cells	13	5	100	6
Num. of cells	77	52	1712	64
Num. of EPICs	10	3	150	30

Table: Data descriptions.

Experiments

	Data	SKCM (AUC)		PD (AUC)		Bike Sharing (RMSE)		Cal Housing (RMSE)	
	Model	Train	Test	Train	Test	Train	Test	Train	Test
not interpretable	MOE	.973	.756	.891	.815	47.6	52.8	.684	.687
	CWM	-	-	-	-	38.5	109.2	.369	.623
	RF	.997	.688	.997	.794	43.8	52.2	.421	.554
	SVR	.764	.667	.756	.728	145.6	148.1	.671	.676
	MARS	-	-	-	-	141.1	140.1	.629	.640
	MLP	.987	**.777**	.975	**.833**	40.1	**47.6**	.503	**.517**
interpretable	LR	.826	.702	.880	.789	141.1	140.3	.723	.727
	SAR	-	-	-	-	-	-	.655	.652
	GA^{1}M	-	-	-	-	99.4	101.1	.613	.640
	MLM-cell	.948	.728	1.000	**.860**	52.7	**60.9**	.560	**.570**
	MLM-EPIC	.861	**.742**	.975	.851	62.8	66.7	.569	.584

Table: Prediction accuracy. **MOE**: Mixture of Experts, **CWM**: Cluster Weighted Modeling, **MARS**: Multivariate Adaptive Regression Splines, **LR**: Linear Regression, **SAR**: Spatial AutoRegression, **GAM**: Generalized Additive Model.

Experiments: Bike Sharing Rentals - Reg. Coef.

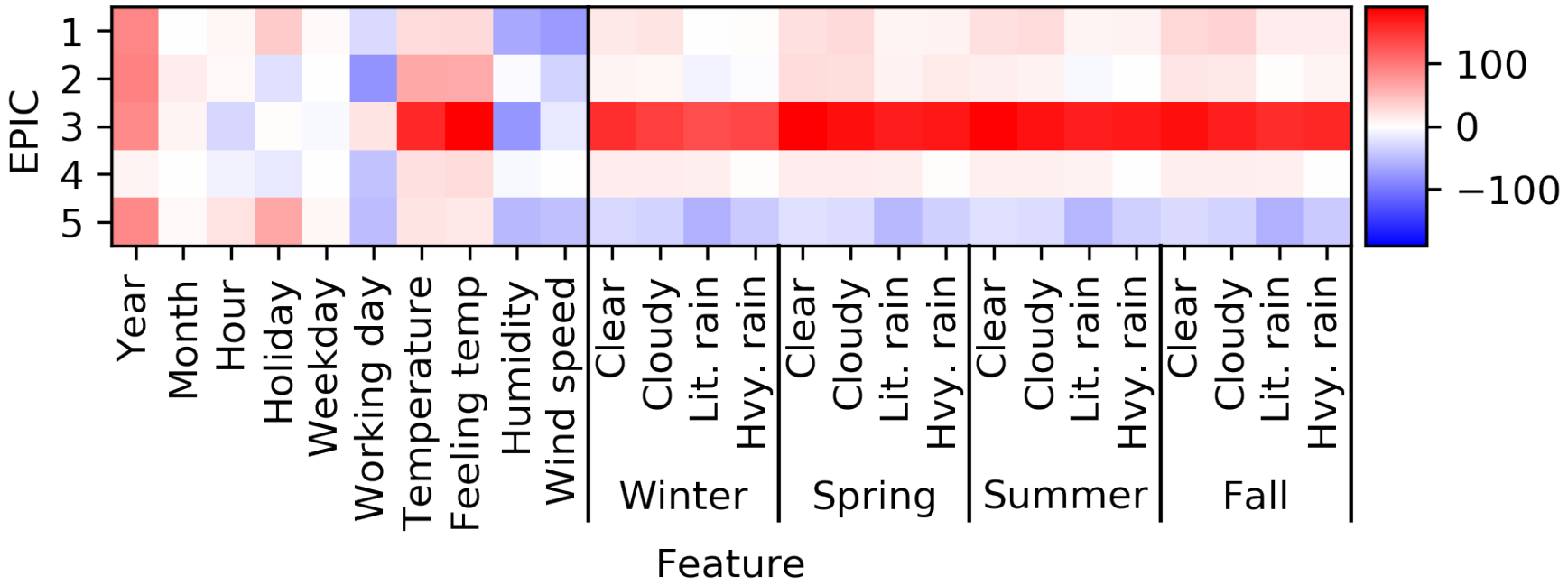


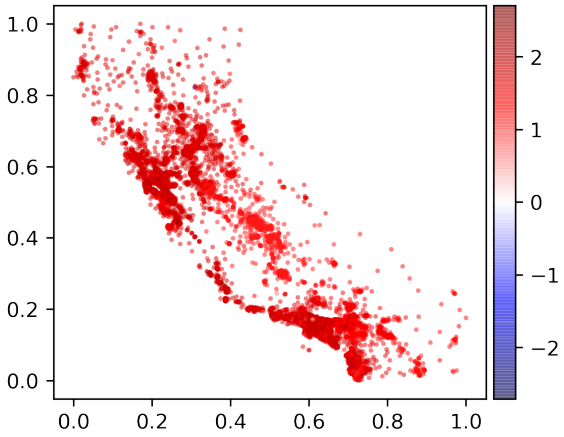
Figure: The linear mixture model regression coefficients $\{\hat{\beta}_j | \mathbf{x} \in \mathcal{P}_j\}_{j=1}^{\widetilde{J}}$ for the top 5 largest EPICs for bike sharing data.

Experiments: Bike Sharing Rentals - PR

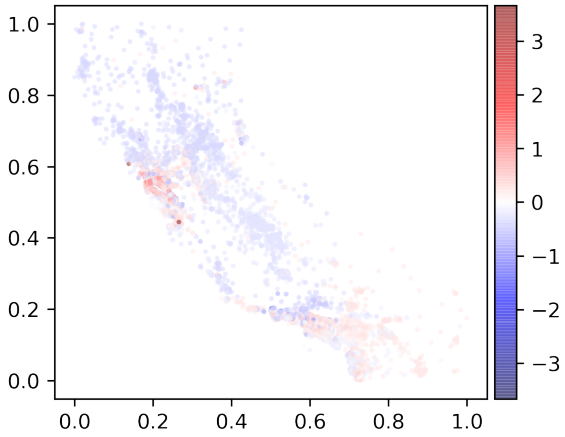
EPIC	Descriptions	Size
1 (883)	Month > 3.5, $11.5 <$ Hour ≤ 14.5, Working day $= 1$	727
2 (488)	Month ≤ 3.5, $9.5 <$ Hour ≤ 14.5, Working day $= 1$, Season $\neq 1$	377
	Year ≤ 2011, Month ≤ 2.5, $10.5 <$ Hour ≤ 14.5, Working day $= 1$, Season $\neq 1$	51
3 (456)	Month > 3.5, $20.5 <$ Hour ≤ 21.5, Working day $= 1$, Season $\neq 3$	166
4 (370)	$9.5 <$ Hour ≤ 11.5, Working day $= 1$, Season $= 1$	182
	$9.5 <$ Hour ≤ 10.5, Working day $= 1$, Season $= 2$, Weather $\neq 2$	88
5 (307)	Year ≤ 2011, Month > 3.5, $13.5 <$ Hour ≤ 15.5, Holiday $\neq 1$, Working day $\neq 1$, Temperature ≤ 0.35, Season $\neq 3$	66
	$11.5 <$ Hour ≤ 14.5, Weekday ≤ 0.5, Working day $\neq 1$, Season $= 1$	59

Table: Explainable conditions for the top 5 largest EPICs for bike sharing data. Numbers in the bracket are the sizes of EPICs in the training data.

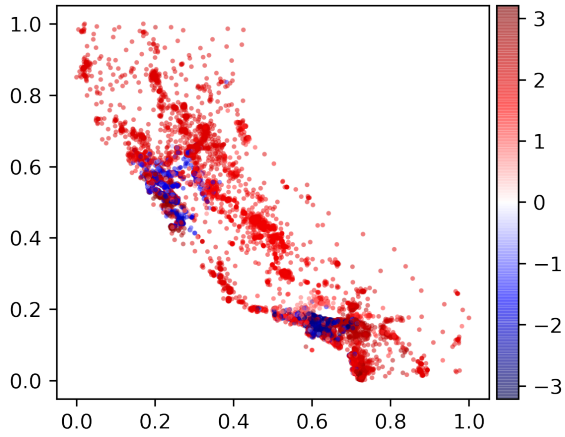
Experiments: Cal Housing Price - Reg. Coef.



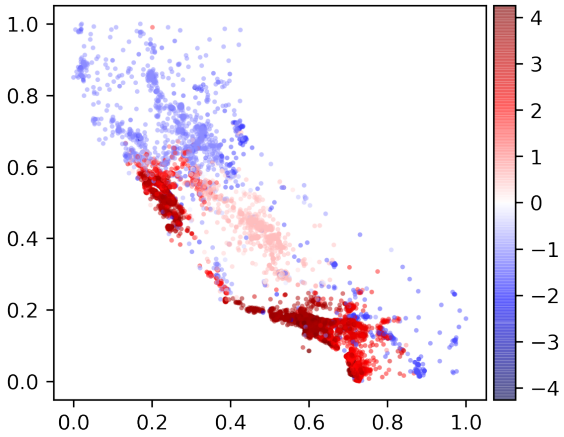
(a) Median income



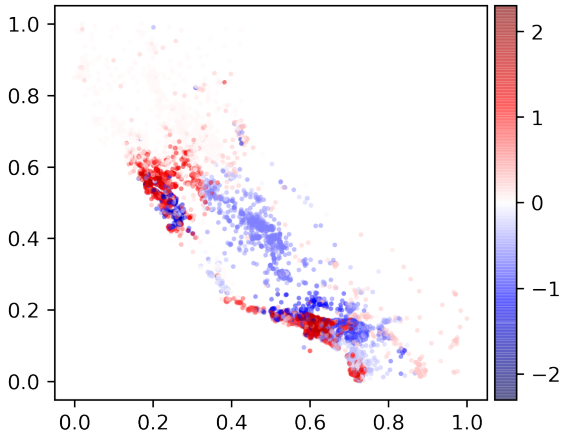
(b) House age



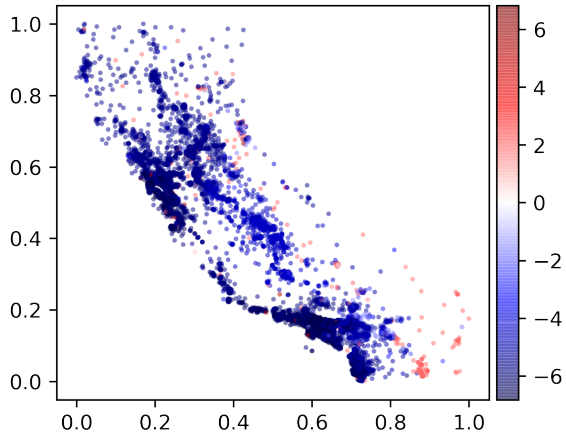
(c) Other rooms



(d) Bedrooms



(e) Population

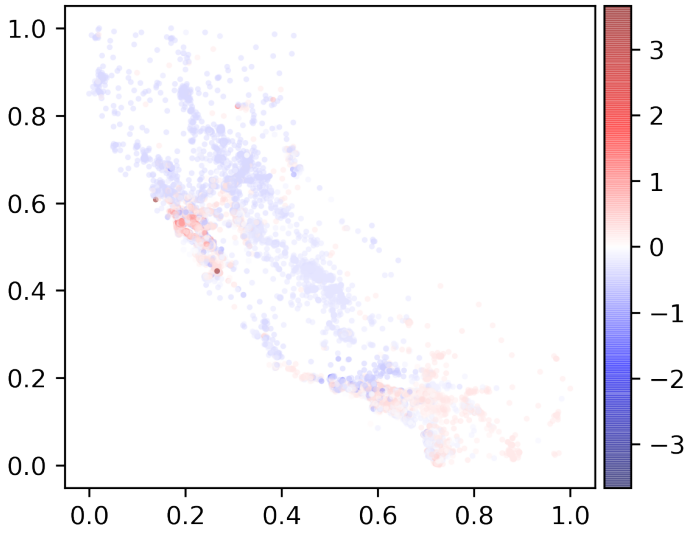


(f) Occupancy

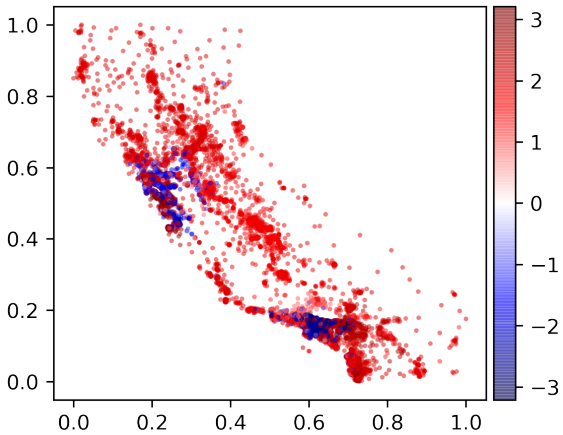
Figure: Log-valued regression coefficients of local linear models plotted on longitude (horizontal axis) and latitude (vertical axis) space.

Experiments: Cal Housing Price - Reg. Coef.

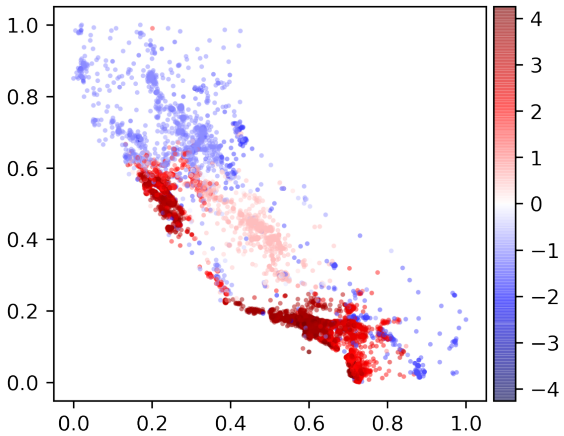
(a) Cal map



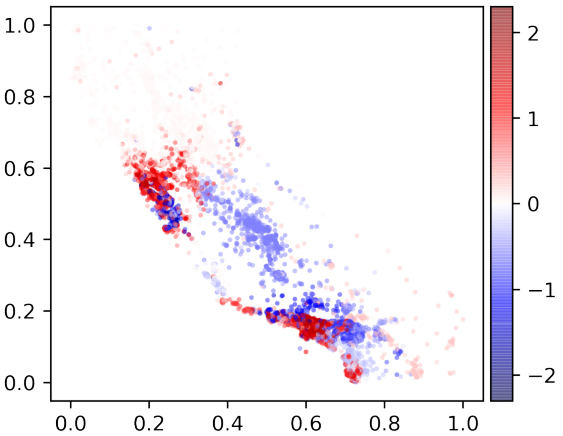
(b) House age



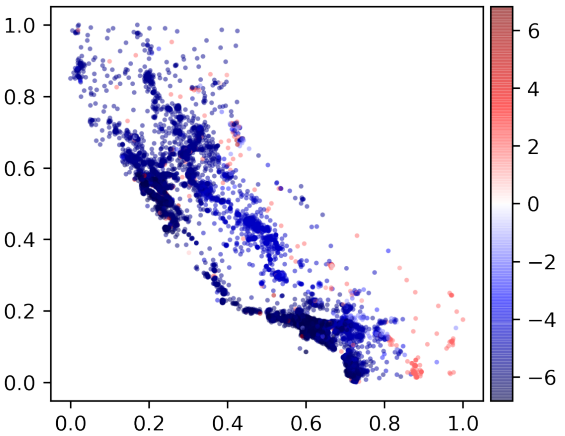
(c) Other rooms



(d) Bedrooms



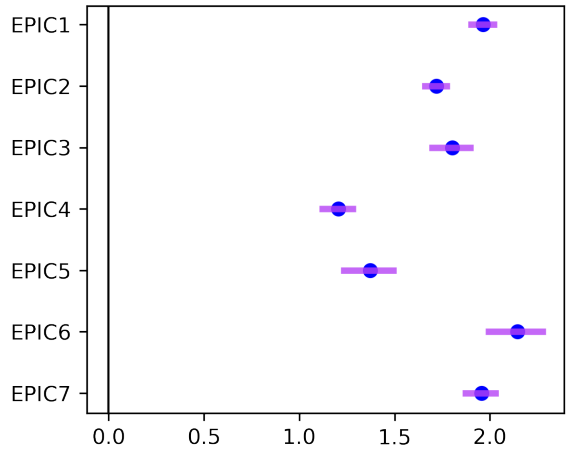
(e) Population



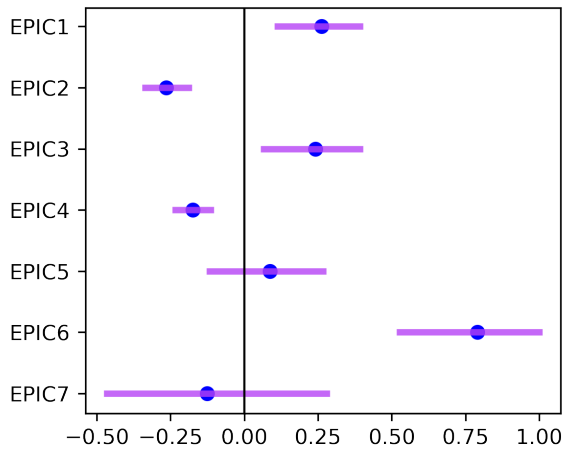
(f) Occupancy

Figure: Log-valued regression coefficients of local linear models plotted on

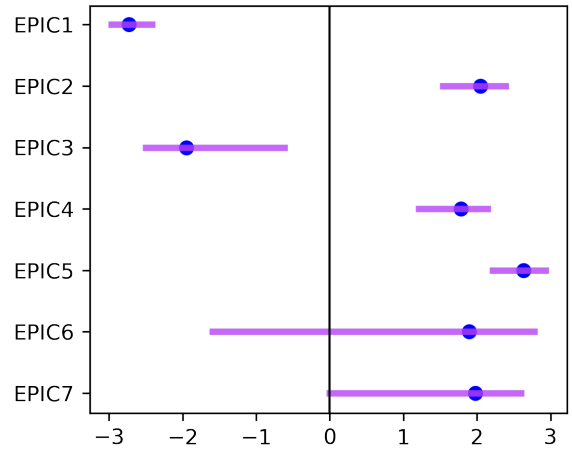
Experiments: Cal Housing Price - Reg. Coef. CI



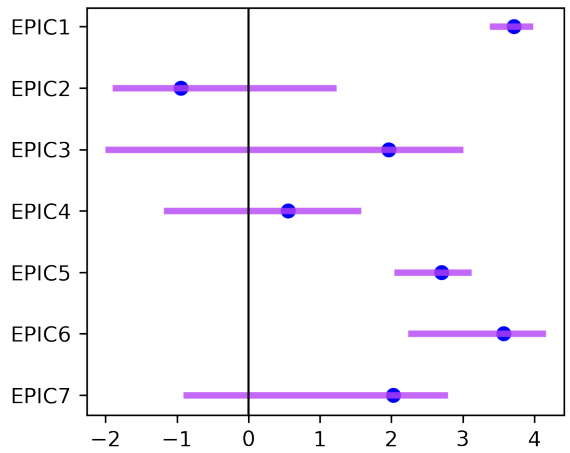
(a) Median income



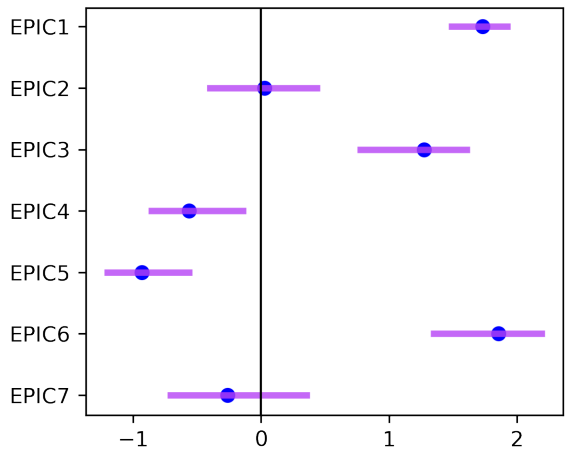
(b) House age



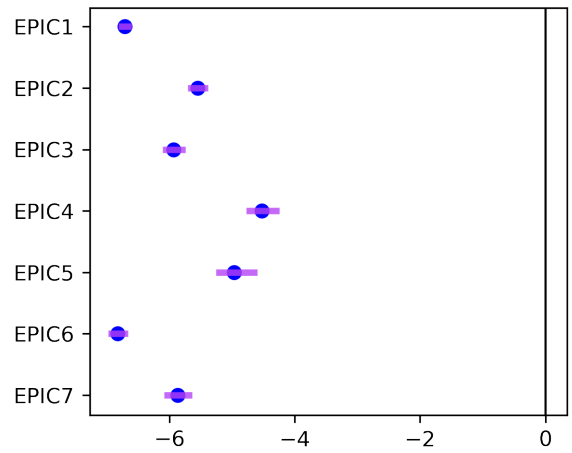
(c) Other rooms



(d) Bedrooms



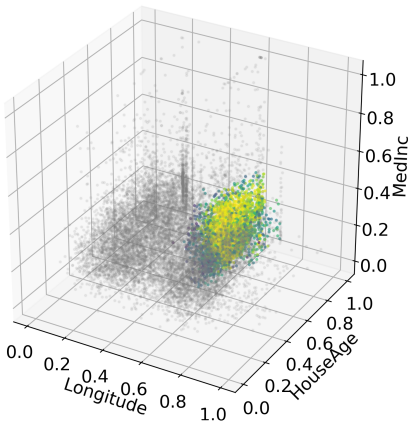
(e) Population



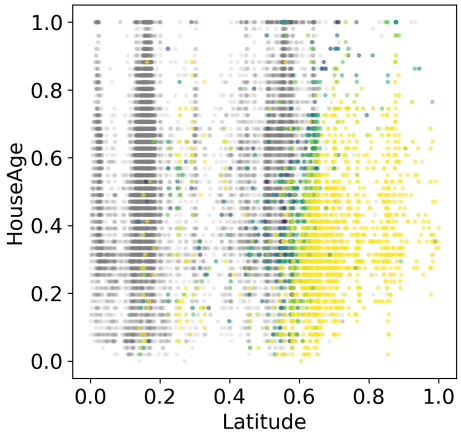
(f) Occupancy

Figure: Log-valued confidence intervals of each variable in the 7 largest EPICs.

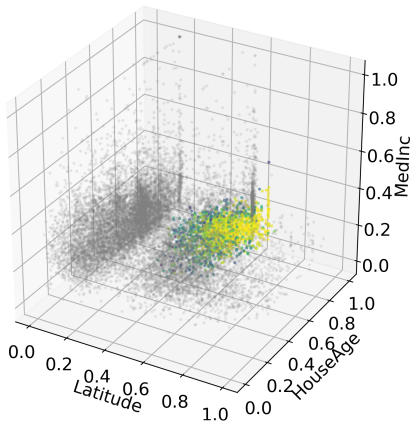
Experiments: Cal Housing Price - LDS



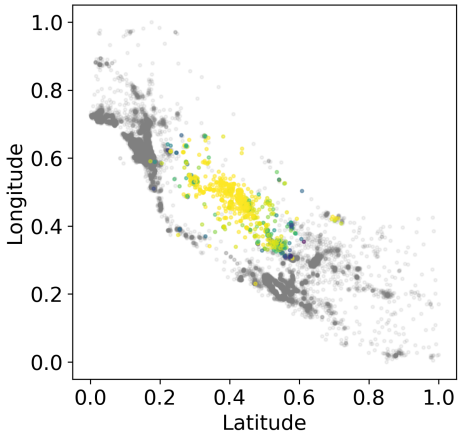
(a) EPIC 1 (0.84)



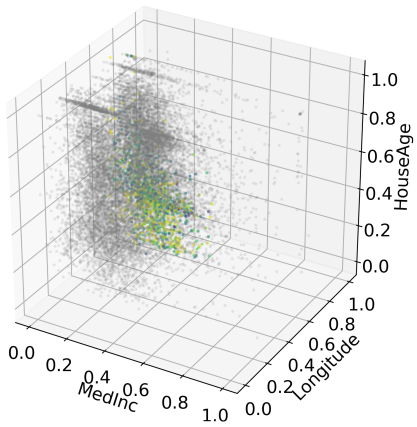
(b) EPIC 2 (0.82)



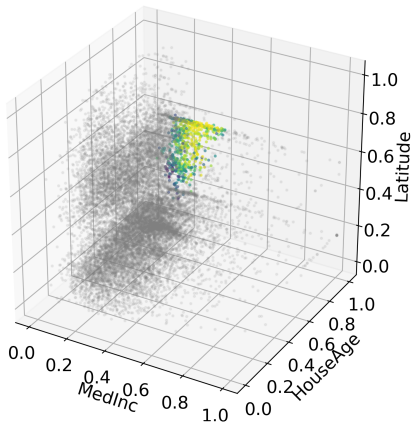
(c) EPIC 3 (0.80)



(d) EPIC 4 (0.88)



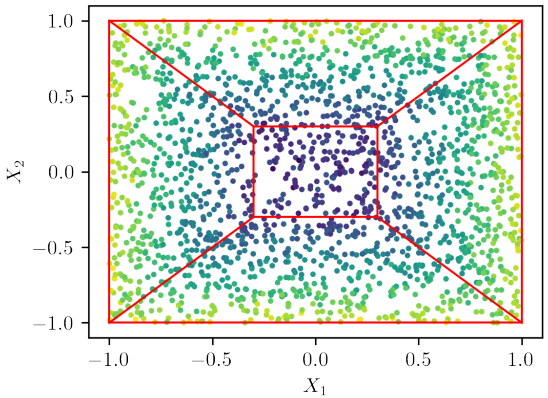
(e) EPIC 5 (0.58)



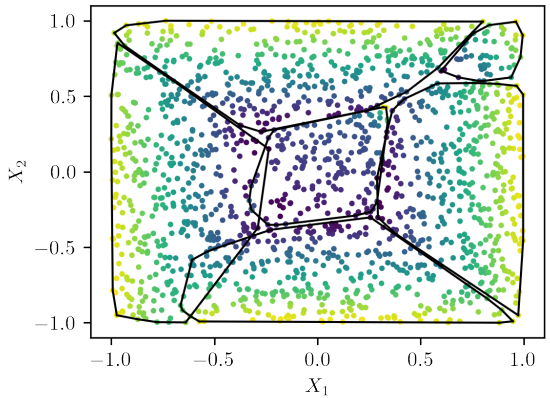
(f) EPIC 6 (0.81)

Figure: Explainable dimensions for the first 6 biggest EPICs. The values in the parentheses are explainable rates of the explainable dimensions for each EPIC.

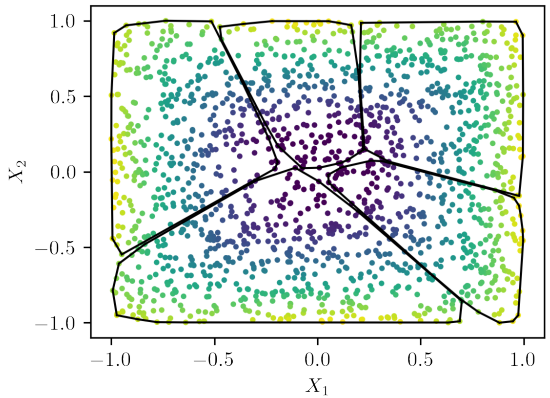
Synthesized Data



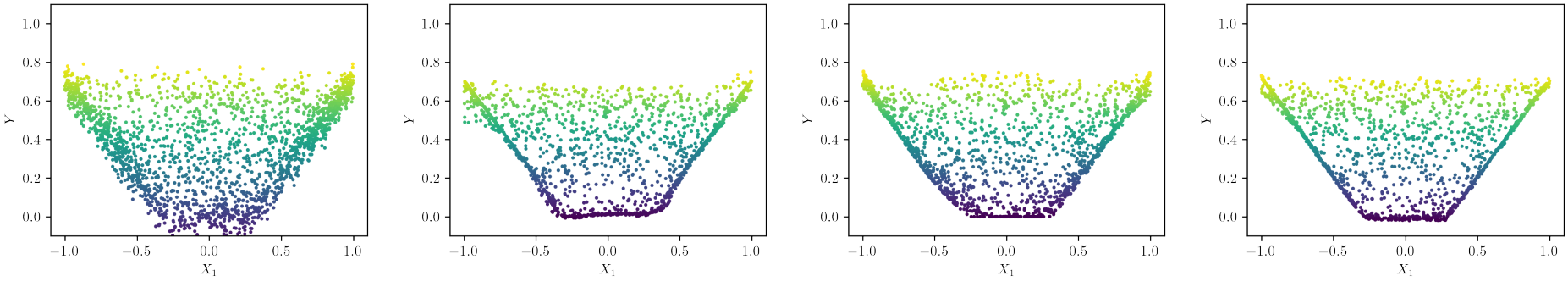
(a) Ground Truth



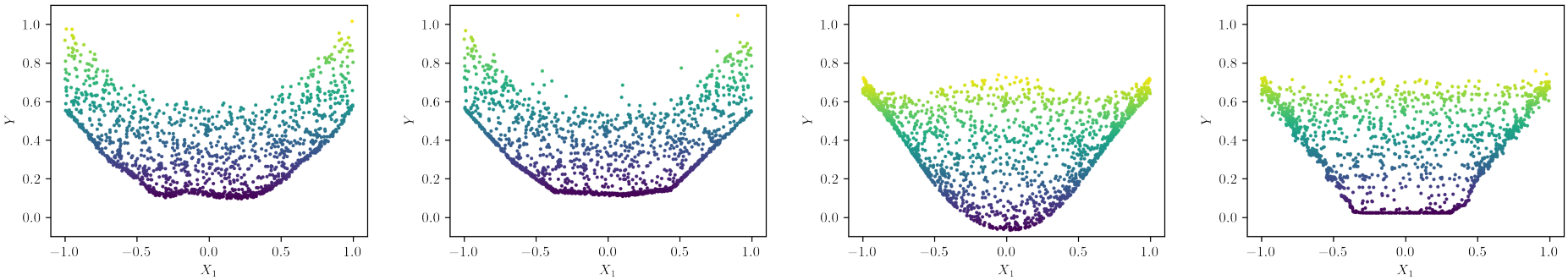
(b) MLM



(c) MOE



(a) Ground Truth (b) MLM (c) MOE (d) DNN



(e) GAM (f) MARS (g) SVR(RBF kernel) (h) RF

Figure: (a-c) Scatter plots on the two explaining variables, X_1 and X_2, with ground truth groups or convex hulls of estimated clusters for each model. (d-k) Scatter plots on the predicted values, $\hat{Y}$, and an explaining variable, X_1, for each model.