Carnegie Mellon University

ICPR 2020
25th INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION
Milan, Italy 10 | 15 January 2021

Hierarchical Routing Mixture of Experts

Wenbo Zhao, Yang Gao, Shahan Ali Memon, Bhiksha Raj, Rita Singh

Carnegie Mellon University

Outline

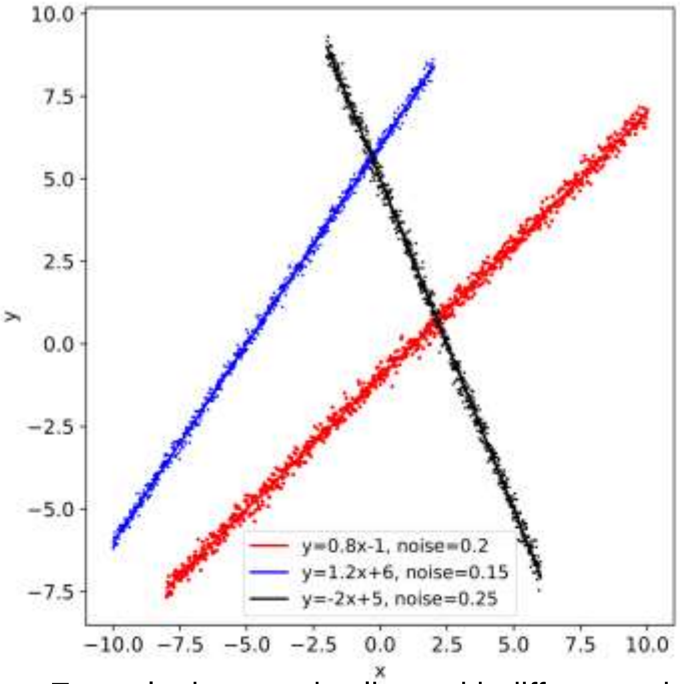
- Motivation & Background
- Hierarchical Routing Mixture of Experts
  - Model
  - Learning Algorithm
- Experiments
- Conclusion

Outline

- **Motivation & Background**
- Hierarchical Routing Mixture of Experts
    - Model
    - Learning Algorithm
- Experiments
- Conclusion

Complex data distributions are challenging for regression tasks

- Complex data distributions
  - E.g., multimodal data
  - Single regression model has high bias

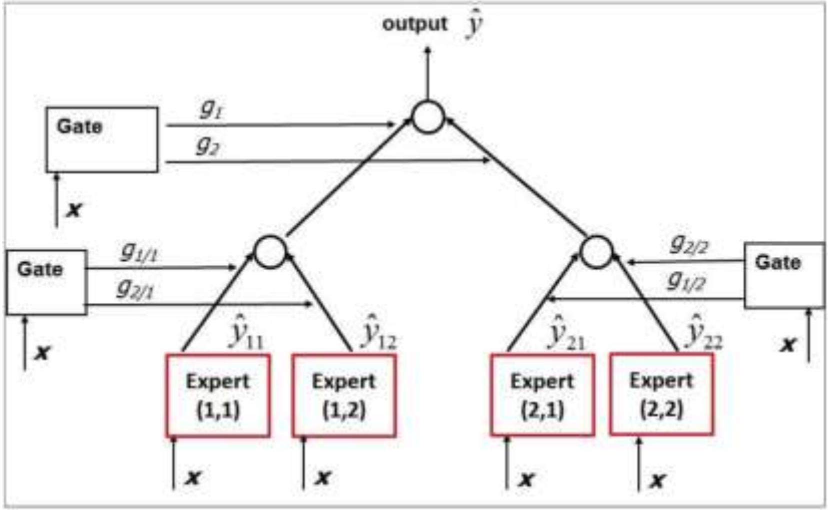


Example: intersecting lines with different noises

Carnegie Mellon University

Regression on complex distributions by divide-and-conquer

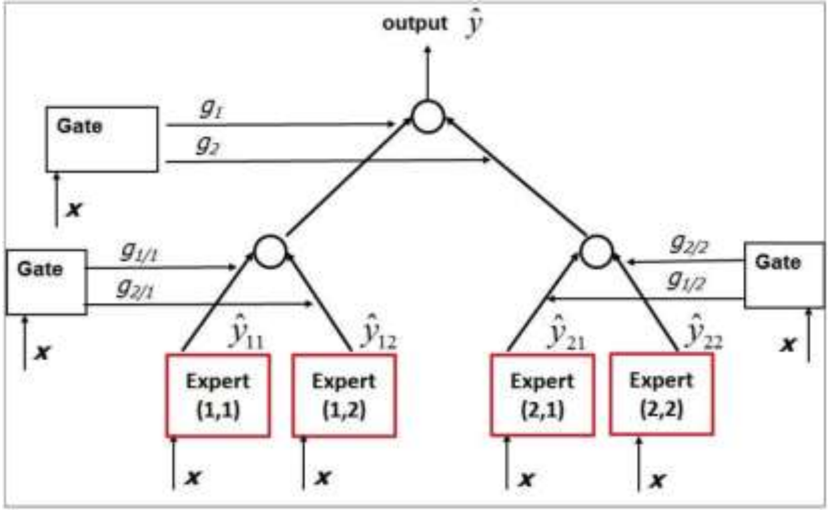
- Conventional divide-and-conquer methods
  - Partition input space
  - Hard-partition: decision trees, random forests
  - Soft-partition: mixture of experts
    - *Probabilistic tree-structured models*
    - *Nodes: gates to partition inputs*
    - *Leaves: experts to local regression*
    - *E.g., HME, HME-GP, HME-SVM*



Hierarchical mixture of experts [1]

Conventional divide-and-conquer methods have shortcomings

- Shortcomings of conventional methods
  - Hard-partition: decision trees, random forests
    1) *Discontinuities*
    2) *High biases*
  - Soft-partition: mixture of experts
    1) *Do not leverage input-output dependency; gate/partition based on assumed distributions*
    2) *Need strong experts*
    3) *Need additional procedures to optimize tree structures*



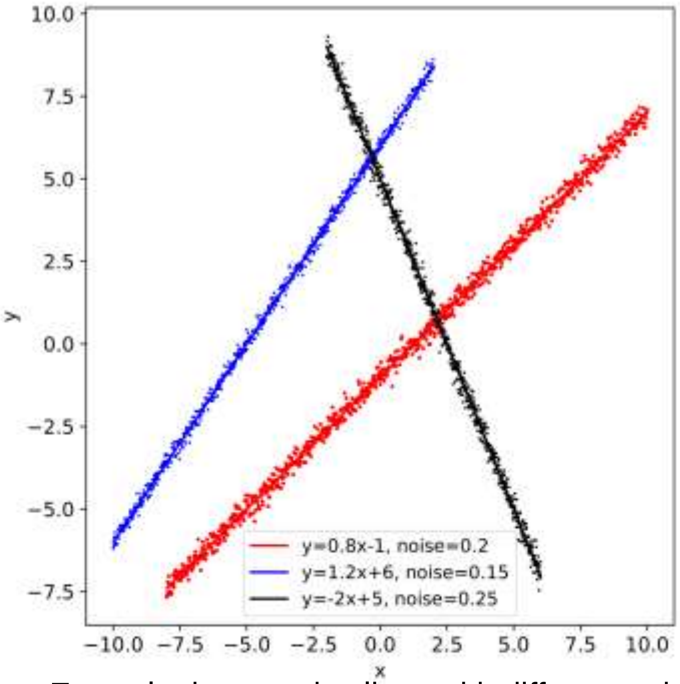
Hierarchical mixture of experts [1]

Carnegie Mellon University

We address conventional methods' shortcomings by joint-partition and optimization

- Joint partition input-output space
  - E.g., different sub-output spaces (y) have different modes (x)
  - Joint partition (x, y) such that each sub-region has a simple mode to enable simple expert

- Joint optimization tree structure and experts



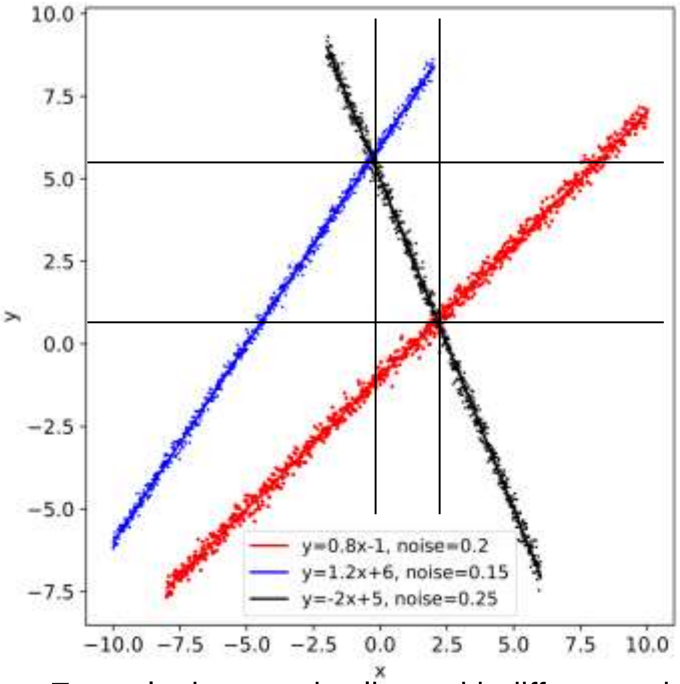
Example: intersecting lines with different noises

Carnegie Mellon University

We address conventional methods' shortcomings by joint-partition and optimization

- Joint partition input-output space
  - E.g., different sub-output spaces (y) have different modes (x)
  - Joint partition (x, y) such that each sub-region has a simple mode to enable simple expert

- Joint optimization tree structure and experts
  - No need for additional structure optimization



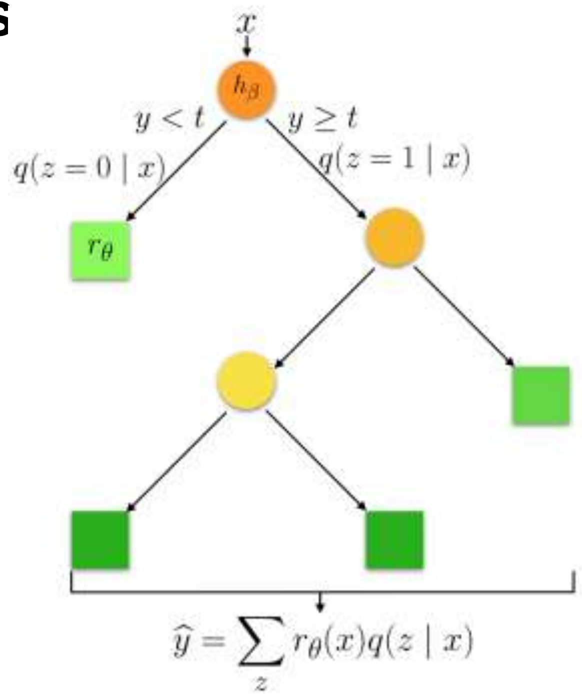
Example: intersecting lines with different noises

Outline

- Motivation & Background
- **Hierarchical Routing Mixture of Experts**
    - Model
    - Learning Algorithm
- Experiments
- Conclusion

Hierarchical routing mixture of experts (HRME) has classifier nodes and regressor leaves

- Binary tree
  - Node: binary classifier
    - *Classify by separateness of modes*
    - *Soft-partition by probabilistic class assignment*
    - *Hierarchical partition input-output space*
    - *Resulting sub-region has simple mode, ideally unimodal*
  - Leaf: simple regressor
    - *Each sub-region has a regressor*



HRME model

Hierarchical routing mixture of experts (HRME) makes probabilistic inference

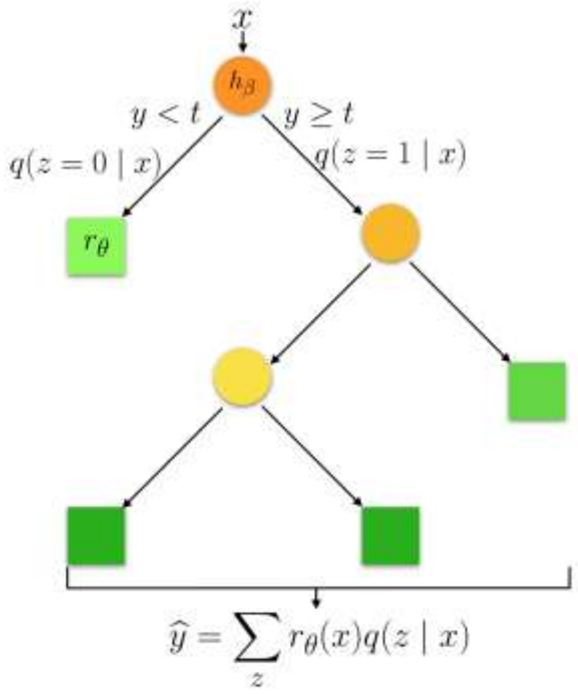
- Probabilistic inference for data $(\boldsymbol{x}, y), \boldsymbol{x} \in \mathbb{R}^d, y \in \mathbb{R}$
  - Introduce a threshold $t, y = 0\ if\ y < t\ otherwise\ y = 1$
  - Each node n_i carries a classifie $h_{\beta^*_{n_i}} : \boldsymbol{x} \mapsto \{n_{i+1}, n_{i+2}\}$
  - Introduce a binary-valued random variable z_{n_i}

    1: assign to n_i, 0: not assign
  - Likelihood of assign a data $\boldsymbol{x}$ to node n_i

    $$q(\mathrm{z}_{n_i} \mid \boldsymbol{x}) \equiv q(\mathrm{z}_{n_i} = 1 \mid \boldsymbol{x}) \longleftarrow h_{\beta^*_{n_{i-1}}}(\boldsymbol{x})$$
  - Likelihood of assign a data $\boldsymbol{x}$ to leaf l_k

    $$q(\mathrm{z}_{l_k} \mid \boldsymbol{x}) = \prod_{j=1}^{k-1} q(\mathrm{z}_{l_{j+1}} \mid \mathrm{z}_{l_j}, \boldsymbol{x})$$
  - Estimate by expectation of leaf predictions $r_{\theta^*_{l_k}}(\boldsymbol{x})$

    $$\hat{y} = \sum_{l_k \in \text{leaves}} r_{\theta^*_{l_k}}(\boldsymbol{x}) q(\mathrm{z}_{l_k} \mid \boldsymbol{x}) \qquad p(y \mid \mathrm{z}_{l_k}, \boldsymbol{x}) \leftarrow r_{\theta^*_{l_k}}(\boldsymbol{x})$$



HRME model

Carnegie Mellon University

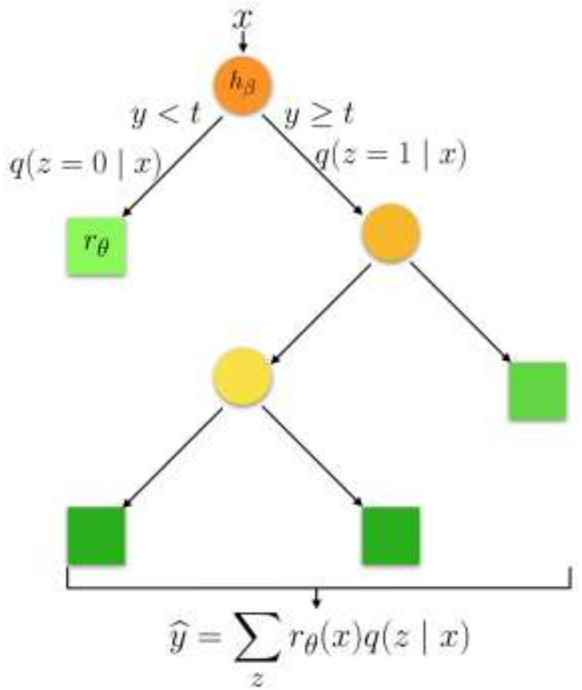
Recursive EM jointly optimizes experts and tree structure

- Recursive Expectation-Maximization algorithm
    - Objective $\max \log p(y \mid \boldsymbol{x}) = \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{p(y, \mathrm{z} \mid \boldsymbol{x})}{q(\mathrm{z} \mid \boldsymbol{x})} + \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{q(\mathrm{z} \mid \boldsymbol{x})}{p(\mathrm{z} \mid y, \boldsymbol{x})},$
    - E-step: compute evidence lower bound (ELBO)

$$Q(p, q) = \sum_{\boldsymbol{x}} \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{p(y, \mathrm{z} \mid \boldsymbol{x})}{q(\mathrm{z} \mid \boldsymbol{x})}$$
$$= \sum_{\boldsymbol{x}} \sum_{\mathrm{z}} q(\mathrm{z} \mid \boldsymbol{x}) \log \frac{p(y \mid \mathrm{z}, \boldsymbol{x}) p(\mathrm{z} \mid \boldsymbol{x})}{q(\mathrm{z} \mid \boldsymbol{x})}$$

    - M-step: optimize partition thresholds and model parameters
    - Done recursively, depth first

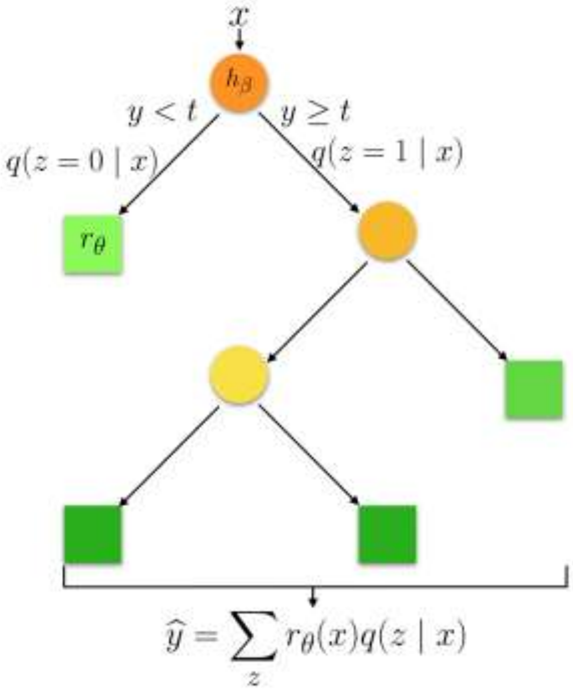
(Details are in the paper)



HRME model

Recursive EM jointly optimizes experts and tree structure

- Recursive Expectation-Maximization algorithm




```
Algorithm 1: Recursive EM Learning of HRME
Input: [data], [root]
Parameter : {t}, classifier parameters, regressor
            parameters
Output: HRME Tree
Function GrowTree(data_list, nodes_per_level)
  for node in nodes_per_level do
    D ← data_list
    node_l, node_r ← GrowSubtree(node)
    for t do
      D_l, D_r ← SplitData(D, t)
      if min(|D_l|,|D_r|) / # of total samples < min_leaf_sample_ratio
       then continue;
      node.TrainClassifier(D, t)
      Propagate conditionals using Equation (3)
      node_l.TrainLeaf(D_l)
      node_r.TrainLeaf(D_r)
      Q ← ComputeQ  using Equation (10)
    end
    if Q > Q* then
      Q* ← Q
      data_list ← [D_l, D_r]
      nodes_per_level ← [node_l, node_r]
      GrowTree(data_list, nodes_per_level)
    else
      Delete the subtree
      continue
    end
  end
```

Outline

- Motivation & Background
- Hierarchical Routing Mixture of Experts
    - Model
    - Learning Algorithm
- **Experiments**
- Conclusion

Experiments

- Data

TABLE I: Dataset Statistics

DATASET	FEATURE DIM	TRAIN	TEST
3-LINES	1	1750	750
HOUSING	13	354	152
CONCRETE	8	721	309
CCPP	4	6697	2871
ENERGY	28	14803	4932
KIN40K	8	10000	30000

- Models
  - HRME
    - *Leaf: linear regression (HRME-LR)*
    - *Leaf: support vector regression (HRME-SVR)*
  - Baselines
    - *Linear regression (LR)*
    - *Support vector regression (SVR)*
    - *Decision trees (DT)*
    - *Random forests (RF)*
    - *Hierarchical mixture of experts (HME)*
    - *Multilayer neural nets (MLP)*

Experiments

- Results

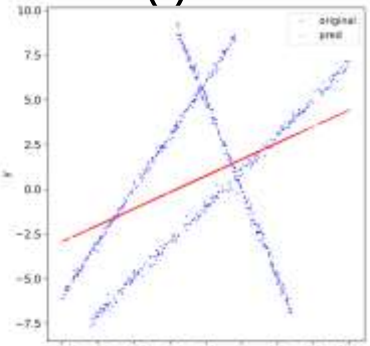
TABLE II: Experiment Results

DATASET	METRIC	LR	SVR	DT	RF	HME	MLP	**HRME**	
								LR	SVR
3-LINES	MAE	3.352	2.006	2.224	2.131	—	1.960	2.337	2.250
	RMSE	4.104	3.173	3.291	3.072	—	2.795	2.885	2.859
HOUSING	MAE	3.651	3.498	2.537	2.103	4.170 [1]	6.711	2.682	3.266
	RMSE	4.911	5.126	3.665	3.043	5.610 [2]	8.535	3.857	4.376
CONCRETE	MAE	8.088	8.013	4.919	3.436	—	5.394	4.121	4.020
	RMSE	10.204	10.772	8.000	4.806	6.250 [3]	6.594	5.664	5.609
CCPP	MAE	3.601	2.746	2.941	2.383	—	4.013	2.965	2.712
	RMSE	4.578	3.856	4.151	3.409	4.100 [4]	5.078	3.951	3.805
ENERGY	MAE	52.075	43.141	43.996	52.002	—	40.521	42.121	40.009
	RMSE	93.564	101.267	99.654	95.558	—	88.191	89.203	87.022
KIN40K	MAE	0.806	0.092	0.592	0.433	—	0.237	0.150	0.071
	RMSE	0.996	0.161	0.773	0.548	0.230 [5]	0.312	0.212	0.114

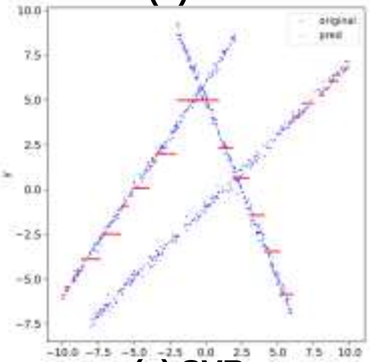
Experiments

- Results: three-line fitting

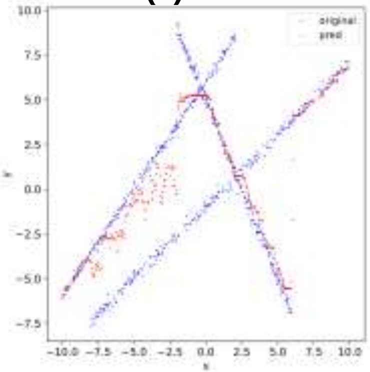
(a) LR



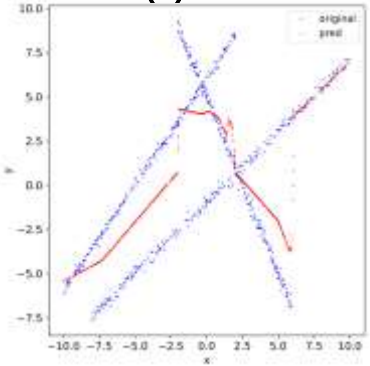
(b) DT



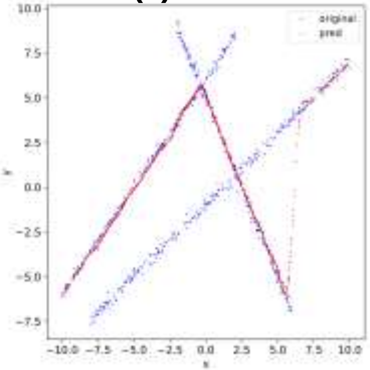
(c) RF

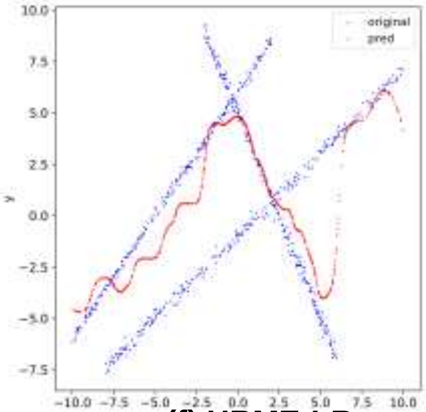


(d) MLP

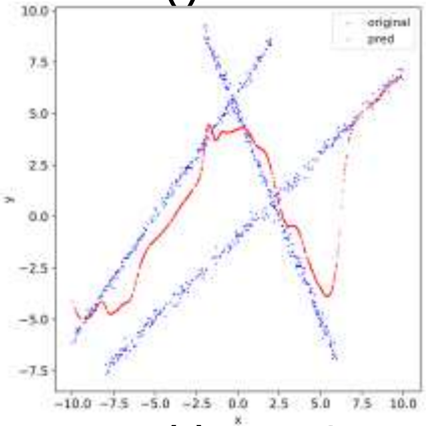


(e) SVR





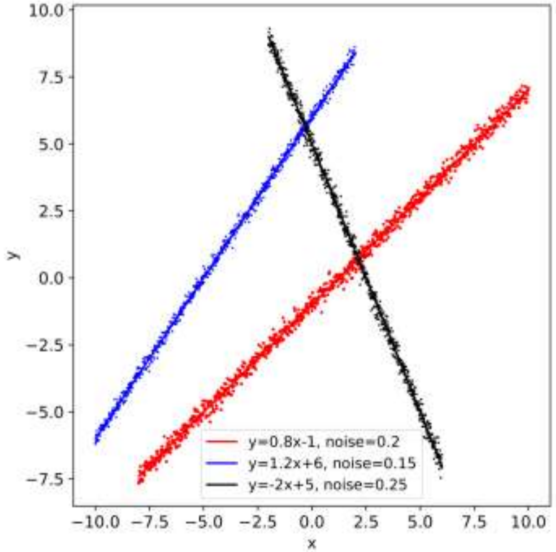
(f) HRME-LR



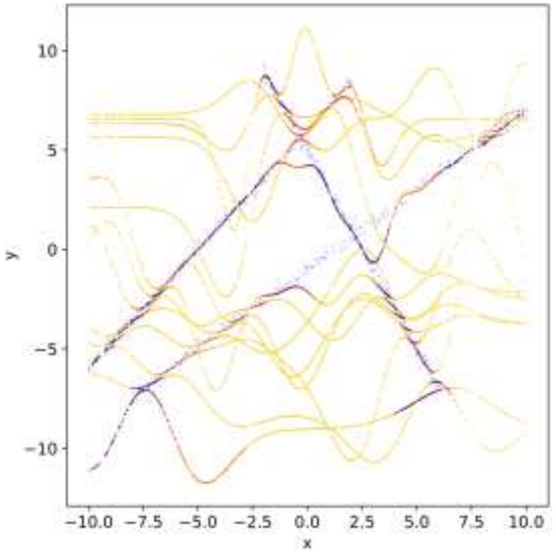
(g) HRME-SVR

Experiments

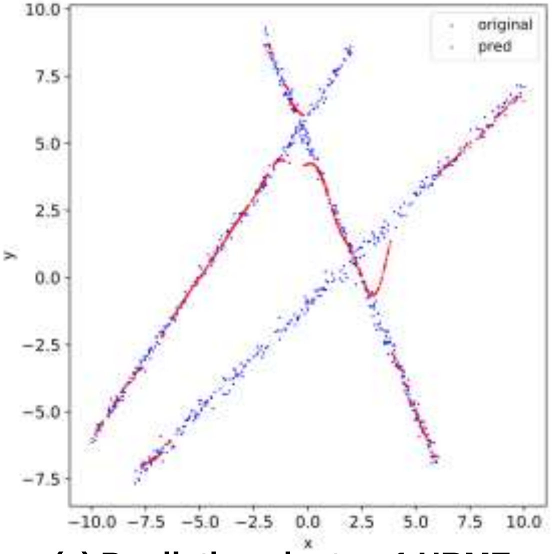
- Results: three-line fitting by experts in HRME



(a) Data



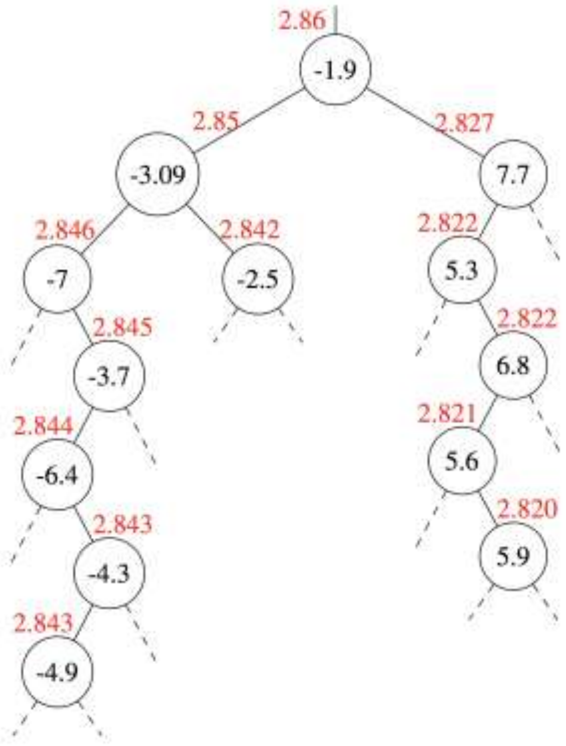
(b) Predictions by HRME experts
Darker color: stronger confidence



(c) Predictions by top-1 HRME experts

Experiments

- Results: HRME tree on three-line data



HRME tree
Node: partition threshold
Edge: regression error

Carnegie Mellon University

Outline

- Motivation & Background
- Hierarchical Routing Mixture of Experts
    - Model
    - Learning Algorithm
- Experiments
- **Conclusion**

Conclusion

- Hierarchical routing mixture of experts (HRME) addresses the difficulty of data partitioning and expert assigning in conventional regression models
- HRME captures natural data hierarchy and routes data to simple regressors for effective predictions
- Probabilistic framework + recursive Expectation-Maximization (EM) algorithm to optimize both tree structure and expert models
- Comprehensive experiments validate effectiveness
- HRME properties
    - Convergence: $\mathcal{O}(k^{-2/d})$ in the L_p norm
    - Complexity: $\mathcal{O}(n^3\epsilon^d + dn^2\epsilon^{d/2})$
    - Consistency: yes
    - Identifiability: yes

Carnegie Mellon University

References

[1] Yuksel, S. E., Wilson, J. N., & Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(8), 1177-1193.

[2] Zhao, W., Gao, Y., Memon, S. A., Raj, B., & Singh, R. (2020). Hierarchical routing mixture of experts. *ICPR 2020*.

Carnegie Mellon University